

MASTER'S THESIS

Programmeeromgevingen met Visuele of Tastbare Output; de Effecten op Computational Thinking

Petrus, Michelle

Award date:
2019

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain.
- You may freely distribute the URL identifying the publication in the public portal.

Take down policy

If you believe that this document breaches copyright please contact us at:

pure-support@ou.nl

providing details and we will investigate your claim.

Downloaded from <https://research.ou.nl/> on date: 05. May. 2023

Open Universiteit
www.ou.nl



Programmeeromgevingen met Visuele of Tastbare Output; de
Effecten op Computational Thinking

Programming Environments with Visual or Tangible Output; the
Effects on Computational Thinking

Michelle Petrus

Master Onderwijswetenschappen

Open Universiteit Nederland

Datum: : 26 juli 2019

Begeleiders: Nardie Fanchamps MSc / prof. dr. Marcus Specht

Inhoudsopgave

Samenvatting.....	3
Summary.....	4
Inleiding.....	5
21 ^e eeuwse vaardigheden.....	6
Programmeren en programmeeromgevingen.....	8
Effecten van visuele en tastbare output bij programmeren.....	9
Computational thinking.....	10
Vraagstelling en hypothesen.....	12
Methode.....	13
Ontwerp.....	13
Participanten.....	14
Materialen.....	14
Interventie.....	14
Computational Thinking test.....	15
Rekenraadsels.....	17
Observatie.....	17
Procedure.....	17
Data-analyse.....	19
Resultaten.....	20
Computational thinking skills	20
Debugging.....	21
Algoritmische vaardigheden.....	22
Probleem decompositie.....	22
Rekenraadsels.....	23
Observatie Challenges (sessie 5).....	24
Tijd.....	24
Efficiëntie.....	25
Conclusie en discussie.....	25
Conclusie en discussie.....	25
Beperkingen.....	27
Aanbevelingen.....	28
Referenties.....	29
Bijlage 1 Informatiebrief ouder(s)/verzorger(s).....	35
Bijlage 2 Vragenlijsten voor- en nameting.....	36

Bijlage 3	Interventiesessies.....	77
-----------	-------------------------	----

Programmeeronderwijs binnen Programmeeromgevingen met Visuele of Tastbare Output; de Effecten op Computational Thinking.

Michelle Petrus

Samenvatting

Het ontwikkelen van computational thinking skills stelt kinderen en jongeren in staat om op een andere manier te denken tijdens het oplossen van problemen. Men leert problemen te analyseren vanuit een ander perspectief. Eerder onderzoek heeft zich nog weinig gericht op het vergelijken van verschillende programmeeronderwijs-interventies binnen het primair onderwijs op de ontwikkeling van computational thinking skills. Het doel van deze studie was om inzicht te krijgen in de invloed van visuele of tastbare output van de programmeeromgeving op de ontwikkeling van computational thinking skills bij leerlingen in de leeftijd van 9 tot en met 13 jaar tijdens het leren programmeren. Het onderzoek is uitgevoerd onder 52 leerlingen van een school voor primair onderwijs. Deze 52 leerlingen werden at random toegewezen aan twee onderzoeksgroepen. Door middel van een kwantitatieve voor- en nameting is de ontwikkeling in beeld gebracht. De computational thinking skills werden gemeten door middel van de Computational Thinking test (Román-González, Pérez-González, & Jiménez-Fernández, 2015) en vijf rekenraadsels. Tevens werd, tijdens de laatste challenge, geobserveerd of er een verschil was in de mate van efficiënt oplossen van de deelchallenges en het oplossen binnen de gestelde tijd tussen de verschillende programmeeromgevingen. Om de verschillen op computational thinking skills tussen het werken met een programmeeromgeving met visuele of tastbare output te onderzoeken is er tijdens de interventie gewerkt met het computerprogramma Bomberbot (visuele output) en de meer tastbare Lego EV-3. De groep die heeft gewerkt met de programmeeromgeving met tastbare output heeft gemiddeld een, niet significant ($p = .90$), hogere score op de computational skills ten opzichte van de groep die heeft gewerkt met de programmeeromgeving met visuele output. De resultaten van het huidig onderzoek tonen aan dat er geen significant verschil is op de ontwikkeling van computational thinking skills na het werken met de programmeeromgeving met visuele of tastbare output. Deze resultaten zijn in tegenspraak met eerdere onderzoeken en de vooraf gestelde hypothesen. Dit kan worden verklaard door de relatief kleine onderzoeksgroep.

Keywords: programmeren, computational thinking, debugging, probleemdecompositie, algoritmische vaardigheden en primair onderwijs

Educational Programming within Programming Environments with Visual or Tangible Output; the Effects on Computational Thinking

Michelle Petrus

Summary

The development of computational thinking skills enables children and young people to think differently while solving problems. They learn to analyze problems from a different perspective. Little research has been done on comparing different programming education interventions within primary education on the development of computational thinking skills. The aim of this study was to obtain insight into the influence of visual or tangible output of the program environment on the development of computational thinking skills in students aged 9 to 13 years when learning to program. Current research has been conducted amongst 52 students from primary education. These 52 students were all at random assigned to two research groups. To determine the development, a quantitative pre and post measurement was performed. The computational thinking skills were measured by the Computational Thinking test (Román-González, Pérez-González, & Jiménez-Fernández, 2015) and five math puzzles. During the last challenge, observations were conducted to measure the difference in degree of efficient resolution of the partial challenges and the resolution within the set time between the different programming environments. To investigate the differences in computational thinking skills between working with a programming environment with visual or tangible output, the computer program Bomberbot (visual output) and the more tangible, robotics programming environment Lego EV-3 were used during the intervention. The group that worked with the programming environment with tangible output has on average a, not significant ($p = .90$), higher score on the computational skills compared to the group that worked with the programming environment with visual output. The results of current research show that there is no significant difference in the development of computational thinking skills after working with the programming environment with visual or tangible output. These results are in contradiction with previous studies and the predetermined hypotheses. This can be explained by the relatively small research group.

Keywords: Programming, computational thinking, debugging, problem decomposition, algorithmic skills and primary school

Inleiding

Probleemschets en doel van het onderzoek

Programmeren binnen het primair onderwijs biedt vele rijke ontwikkelingsmogelijkheden voor kinderen (Clements & Gullo, 1984). Het is een betrekkelijk nieuw aandachtsgebied dat vraagt om een gedegen en doordachte implementatie in het onderwijs (Bocconi, Chiocciariello, Dettori, Ferrari, & Engelhardt, 2016). Het aanbod van programmeeromgevingen is groot. Een zorgvuldige inbedding in een onderwijsprogramma (of in het curriculum van de basisschool) vraagt om een zorgvuldige afweging in mogelijkheden en te bereiken doelen (Bocconi et al., 2016). Daarnaast blijkt uit eerder onderzoek dat er een variatie in leeropbrengst en betrokkenheid bij leerlingen optreedt wanneer zij leren programmeren met een programma met een tastbare output of wanneer de output van het programmeren meer in het abstracte, visuele vlak plaatsvindt (Sapounidis, Demetriadis, & Stamelos, 2015).

Leren programmeren en de functionaliteit die ermee gepaard gaat is beschreven als een van de vaardigheden die valt onder computational thinking; een van de prominente aandachtsgebieden binnen de 21^e eeuwse vaardigheden (Kennisset, 2016; Voorwinden, 2016). Computational thinking omvat vaardigheden die zich richten op het procesmatig (her)formuleren van problemen, zodanig dat men door middel van computertechnologie de problemen kan oplossen (SLO, 2015). Echter is er nog geen eenduidige definitie voor het begrip computational thinking (Barr & Stephenson, 2011; Brennan & Resnick, 2012).

Om binnen het primair onderwijs meer inzicht te krijgen in de effecten van het programmeeronderwijs, en welke vorm van output (visuele of tastbare) daarbij de meest prominente opbrengsten genereren, is het interessant om te onderzoeken of er verschillen in leeropbrengst zijn wanneer leerlingen binnen een programmeeromgeving werken met een visuele of met tastbare output om computational thinking skills te verwerven. Voor dit onderzoek is er een selectie gemaakt van drie deelvaardigheden van computational thinking skills, namelijk: algoritmische vaardigheden, debugging en probleemdecompositie.

Uit eerder (review)onderzoek blijkt dat er weinig kwantitatief vergelijkend onderzoek is uitgevoerd naar de leeropbrengsten op het gebied van computational thinking van verschillende programmeeronderwijs-interventies binnen het primair en voortgezet onderwijs (Bocconi et al., 2016; Jeuring, Corbalan, Van Es, Van Leeuwenstein, & Van Montvoort, 2016; Lye & Koh, 2014). Daarnaast is het belangrijk om kinderen al op vroege leeftijd kennis te laten maken met computational thinking (Bocconi et al., 2016). Derhalve zal een kwantitatieve studie worden uitgevoerd onder leerlingen uit het primair onderwijs in de leeftijd van 9 tot en met 13 jaar om deze mogelijke verschillen in leeropbrengst bloot te leggen. In deze studie wordt daarom de volgende hoofd-onderzoeksvraag onderzocht: “Wat is de invloed van visuele of tastbare output van de programmeeromgeving op de

ontwikkeling van computational thinking skills bij leerlingen primair onderwijs in de leeftijd van 9 tot en met 13 jaar en welk effect is daarbij meetbaar op algoritmische vaardigheden, debugging en probleem decompositie?”

Theoretische kader

De 21e eeuw wordt voornamelijk gekenmerkt door de evolutie van technologie en ICT, globalisering en een behoefte aan innovatie, waardoor de noodzaak voor leerlingen om relevante vaardigheden en competenties te ontwikkelen wordt benadrukt (Chalkiadaki, 2018). Tevens zorgt de toename van de digitalisering van communicatie en informatie in de 21^e eeuw ervoor dat de relevante vaardigheden die hiervoor nodig zijn dienen te worden ontwikkeld (Koninklijke Nederlandse Akademie van Wetenschappen, 2012). Iedereen heeft toegang tot nieuwe media (Twitter, YouTube, Blogs, WhatsApp, enz.), maar ook tot internetbankieren en het opzoeken van informatie via internet. Door deze toepassingen wordt het eenvoudiger om over de grenzen te communiceren en samen te werken. Deze ontwikkelingen zorgen ervoor dat de samenleving verandert van een industriële samenleving naar een dynamische kenniseconomie (Anderson, 2008; van Laar, van Deursen, van Dijk, & de Haan, 2017) en het accent verschuift van het produceren van standaardproducten naar het ontwikkelen en circuleren van kennis (Thijs, Fisser, & van der Hoeven, 2014). Dit wordt onderschreven door Trilling en Fadel (2009) die spreken over een innovatie-economie. Hierin draait het om het genereren van nieuwe ideeën door gebruik te maken van creativiteit, samenwerking ondernemerschap en technologische toepassingen. Deze veranderingen zorgen ervoor dat de gehele leefomgeving waarin mensen functioneren verandert. Wanneer de samenleving verandert zal het onderwijs zich hierop moeten aanpassen. Dit heeft consequenties voor het huidige onderwijs. Staatsecretaris Dekker (OCW) reageerde als volgt op het eindadvies 2032: “Het werken en leren in de digitale wereld behoort tot de kern van toekomstgericht onderwijs [...] Dat betekent dat leerlingen ICT-basiskennis opbouwen, informatievaardigheid ontwikkelen, mediawijs worden, en leren begrijpen hoe informatietechnologie werkt. Dit betreft niet alleen het gebruik van computers en ICT als consument, maar ook als producent” (Kennisset, 2016).

21^e eeuwse vaardigheden

Unesco (2008) en de Europese Unie (2006) hebben verschillende onderzoeken uitgevoerd om in kaart te brengen welke vaardigheden nodig zijn om aan te kunnen sluiten bij de ontwikkelingen in de samenleving. Deze vaardigheden worden omschreven als 21^e eeuwse vaardigheden. Deze 21^e eeuwse vaardigheden worden omschreven als generieke vaardigheden (niet direct gekoppeld aan een specifiek vakgebied, maar relevant op vele gebieden) en daaraan te koppelen kennis, inzicht en houdingen in verband met hogere orde vaardigheden (Gordon et al., 2009; Thijs et al., 2014). Deze hogere orde vaardigheden geven het vermogen weer om te kunnen omgaan met complexe problemen en

onvoorspelbare situaties uit de 21^e eeuwse samenleving (Gordon et al., 2009; Thijs et al., 2014). Doorgaans omvatten de 21^e eeuwse vaardigheden: samenwerking, communicatie, digitale geletterdheid, burgerschap, probleemoplossende vaardigheden, kritisch denken, creativiteit en productiviteit (Voogt & Roblin, 2012).

Kennisnet heeft deze 21^e eeuwse vaardigheden verwerkt in een model voor scholen, welke bestaat uit zeven competenties die, naast de kernvakken taal en rekenen, zouden moeten bijdragen aan de betrokkenheid, ondernemendheid en nieuwsgierigheid van leerlingen. Deze zeven competenties bestaan uit: samenwerken, sociale en culturele vaardigheden, communiceren, kritisch denken, creativiteit, ICT-geletterdheid en probleemoplossend vermogen. Dit model is daarna in samenwerking met SLO (Stichting Leerplan Ontwikkeling) in 2016 aangepast (Voorwinden, 2016). Deze aanpassing heeft ertoe geleid dat digitale geletterdheid niet meer als een zelfstandige vaardigheid voorkomt. Digitale geletterdheid is uitgesplitst in vier zelfstandige vaardigheden: ICT-basisvaardigheden (het veilig kunnen benoemen en omgaan met computers en computernetwerken), mediawijsheid (kennis, vaardigheden en mentaliteit om bewust en kritisch om te gaan met media), informatievaardigheden (het omgaan met grote hoeveelheden informatie) en computational thinking (zie figuur 1). In het model wordt door middel van een stippellijnen tussen de vaardigheden aangegeven dat de vaardigheden elkaar overlappen (Voorwinden, 2016).



Figuur 1. Model voor 21^e eeuwse vaardigheden zoals het is ontwikkeld door SLO en Kennisnet

Programmeren en programmeeromgevingen

In 2014 heeft de Tweede Kamer een motie aangenomen waarin wordt gevraagd om binnen het primair onderwijs onderzoek te doen naar hoe digitale geletterdheid middels programmeren vormgegeven kan worden. De PO-Raad (sectororganisatie voor het primair onderwijs) heeft in samenwerking met Kennisnet op 25 mei 2016 een nieuwe leerlijn programmeren ontwikkeld die is gepubliceerd door Stichting Leerplan Ontwikkeling (SLO). Deze leerlijn richt zich op het ontwikkelen van computational thinking skills door middel van programmeren.

Binnen het onderwijs wordt gebruik gemaakt van verschillende programmeeromgevingen om leerlingen te leren programmeren (Jeuring et al., 2016). Een programmeeromgeving is de software waarmee het programma tot stand kan worden gebracht, hierbij dient het programmeren als middel om een doel te bereiken (Kelleher, Pausch, & Kiesler, 2007). Het doel is over het algemeen het automatiseren van systemen.

De verschillende programmeeromgevingen kunnen worden onderverdeeld in tekstueel programmeren, visueel programmeren en unplugged programmeren. Ten eerste het tekstueel programmeren, wat ook wel wordt gezien als de traditionele manier van programmeren (Jeuring et al., 2016). Hierbij kan gedacht worden aan programmeertalen zoals Java, Scheme of Python. Binnen deze programmeeromgevingen dient de leerling kennis te hebben van de programmeertaal en de specifieke commando's (Lindberg, Laine, & Haaranen, 2018). Ten tweede het visueel programmeren. Binnen deze programmeeromgeving programmeert de leerling door middel van het slepen en aan elkaar klikken van blokken op pictogrambasis (Lindberg et al., 2018), waarbij deze blokken beïnvloedbare variabelen bevatten. Hierbij is in vergelijking met het tekstueel programmeren geen kennis van het schrijven in codes/programmeertaal vereist en is het over het algemeen onmogelijk om een grammaticale fout te maken (Lindberg et al., 2018). Te denken valt aan programma's zoals Scratch, Bomberbot, Lego Mindstorms en Lego EV-3. Binnen dergelijke programmeeromgevingen kan men een onderscheid aanbrengen tussen een visuele output en een tastbare output (Sapounidis et al., 2015). Een visuele programmeeromgeving wordt gekenmerkt doordat deze puur en alleen ten uitvoer wordt gebracht op het beeldscherm van een computer of tablet (Sapounidis et al., 2015). Scratch of Bomberbot zijn voorbeelden van dergelijke programmeeromgevingen. Programmeeromgevingen met tastbare output worden gekenmerkt doordat hiermee fysiek tastbare objecten via een computerprogramma worden geprogrammeerd (Sapounidis et al., 2015). Lego EV-3 en Arduino zijn hiervan twee voorbeelden. Ten derde bestaat het unplugged programmeren. Hierbij wordt geen gebruik gemaakt van een computer. Bij het unplugged programmeren wordt er bijvoorbeeld gebruik gemaakt van fysieke materialen, blokken en/of kaarten waarop commando's worden beschreven die uitgevoerd dienen te worden (Jeuring et. al., 2016).

Effecten van visuele en tastbare output bij programmeren

Programmeeromgevingen met visuele of tastbare output blijken beiden effectief te zijn tijdens het leren programmeren (Goldstone & Son, 2005). Waar het gebruik van tastbare output vooral leidt tot verbeterende taakprestaties, leidt visuele output voornamelijk tot een betere overdracht van het leerproces (Bassok & Holyoak, 1989).

Wanneer leerlingen leren programmeren middels een programmeeromgeving met visuele output zal dit bij de leerlingen het logische begrip bevorderen (Barmby, Harries, Higgins & Suggate, 2007). Tevens zal door middel van het programmeren met een programmeeromgeving met visuele output de focus op de essentie worden bevorderd waardoor de cognitieve belasting wordt verlaagd (Hsin & Paas, 2015). De leerling heeft hierdoor meer cognitieve ruimte voor het construeren van de mentale representatie (Hsin & Paas, 2015). Daarnaast zal het programmeren met visuele output leerlingen stimuleren de koppeling te maken tussen informeel begrip en de meer formele symbolische representaties van wiskundige concepten (Okita, 2014).

In tegenstelling tot het programmeren met visuele output zullen leerlingen de ruimtelijke representaties makkelijker begrepen door de haptische perceptie (Skulmowski, Pradel, Kühnert, Brunnett & Rey, 2016). Tevens zal de leerling mogelijk beter begrijpend hoe onderliggende programmeerroutines werken (Papert, 1980). Door tastbare output hoeft de leerling zich geen abstracte voorstelling te maken van het resultaat van het programmeren waardoor de cognitieve belasting wordt verlaagd. Hierdoor is er meer cognitieve ruimte om de onderliggende programmeerroutines beter te begrijpen (Skulmowski et al., 2016). Een tastbare output van programmeeromgevingen stimuleert het aanraken, manipuleren, onderzoeken en testen waardoor het leren van kinderen effectiever en natuurlijker wordt ondersteund door middel van concrete objecten die relevant zijn voor de taak (Ishii & Ullmer, 1997; Ullmer & Ishii, 2000). Dit wordt onderschreven door onder andere de cognitieve leertheorie van Piaget (Khandelwal & Mazelek, 2007). Ook blijkt programmeren met tastbare output te leiden tot meer betrokkenheid en reflectie op het leren en een hogere mate van samenwerkend leren dan in vergelijking bij het werken met programmeeromgevingen met alleen visuele output (Rogers, Scaife, Gabrielli, Smith & Harris, 2002; Suzuki & Kato, 1995). Dit is te verklaren doordat een programmeeromgeving met tastbare output het mogelijk maakt om gelijktijdig met een ander te communiceren over de output. Beide personen beschikken gelijktijdig over dezelfde waarneming door middel van de tastbare output, hierdoor wordt de controle een gedeelde verantwoordelijkheid van de samenwerkende leerlingen (Zuckerman, Arida & Resnick, 2005). Er is echter nog weinig vergelijkend onderzoek uitgevoerd tussen de verschillen in opbrengst bij een programmeeromgeving met een visuele of tastbare output op de essentiële elementen binnen een programmeeromgeving voor het ondersteunen van leeractiviteiten (Marshall, 2007) waaronder de ontwikkeling van computational thinking skills (Bocconi et al., 2016; Jeuring et al., 2016; Lye & Koh, 2014).

Computational thinking

Programmeren is een manier om computational thinking skills te verwerven en verder (door) te ontwikkelen (Grover & Pea, 2013). Programmeren zorgt ervoor dat onderdelen van computational thinking concreet worden gemaakt (Bocconi et al., 2016). Voor het begrip computational thinking zijn meerdere definities in omloop die nog niet geheel tot een uniforme duidelijkheid leiden (Bocconi et al., 2016). Computational thinking skills kunnen worden ontwikkeld door programmeren. De vaardigheden probleemanalyse en decompositie, die vallen onder computational thinking, vinden plaats voordat men begint met programmeren (Bocconi et al., 2016; Wing 2006). Wing (2008) definieert computational thinking als een aanpak voor het oplossen van problemen, het ontwerpen van systemen en het begrijpen van menselijk gedrag dat is gebaseerd op concepten welke fundamenteel zijn voor computing (het gebruik of de werking van computers). Shute en collega's (2017) beschrijven computational thinking als de conceptuele basis die nodig is om problemen effectief en efficiënt op te lossen dat wil zeggen algoritmisch, met of zonder de hulp van computers. Hierbij dienen oplossingen gegenereerd te worden die in verschillende contexten te gebruiken zijn (Shute, Sun, & Asbell-Clarke, 2017). Lye en Koh (2014) voegen hieraan toe dat computational thinking skills kunnen worden ontwikkeld door middel van het uitvoeren van programmeeractiviteiten. SLO omschrijft het concept computational thinking als geheel van vaardigheden die zich richt op het procesmatig (her)formuleren van problemen, zodanig dat men door middel van computertechnologie dergelijke problemen kan oplossen (SLO, 2015).

Verschillende onderzoekers hebben zich afgelopen decennia gericht op het onderzoeken van de ontwikkeling van computational thinking skills (Berland & Lee, 2011; Bocconi et al., 2016; Dierbach et al., 2011; Settle, Perkoviæ, Hwang, & Jones, 2010; Wing, 2018). Echter ontbreekt het aan een eenduidig en allesomvattend instrument om de (ontwikkeling van) computational thinking skills te meten bij kinderen. Dit zou mede kunnen komen door de niet eenduidige definitie van computational thinking. Doordat er nog geen volledig eenduidige definitie voor computational thinking gehanteerd wordt, is er een variatie in de karakteristieken en vaardigheden die computational thinking skills genoemd worden. In verschillende onderzoeken worden dan ook tal van suggesties gedaan over de vaardigheden die vallen onder computational thinking (Bocconi et al., 2016). Waar Wing (2008) computational thinking skills vat onder abstractie, generalisatie, decompositie, ontwerp van algoritmes en scheiding van punten van zorg, beschrijven Berland en Lee (2011), Dierbach en collega's (2011) en Settle en collega's (2010) computational thinking skills als abstractie, automatisering, simulatie, evaluatie, algoritmebouw, conditionele logica, debuggen, decompositie, probleemanalyse, gedistribueerde computerverwerking en effectief teamwork. De overlap in de verschillende onderzoeken op het gebied van de computational thinking skills ligt in: abstractie, algoritmisch

denken, automatisering, debugging, decompositie en generalisatie (Bocconi et al., 2016). De definitie van deze items, welke in huidig onderzoek worden gehanteerd, wordt weergegeven in tabel 1.

Tabel 1

Overlappende Computational Thinking Skills en Definities

Computational thinking skills	Definitie
Abstractie	Abstractie is het begrijpelijker maken door onnodige details te verminderen. De vaardigheid in abstractie zit in het kiezen van het juiste detail verbergen, zodat het probleem gemakkelijker wordt, zonder iets te verliezen dat belangrijk is (Csizmadia et al., 2015).
Algoritmisch denken	Algoritmisch denken is een manier om tot een oplossing te komen door middel van een duidelijke set van stappen (Csizmadia et al., 2015).
Automatisering	Automatisering is een arbeidsbesparingsproces waarbij een computer wordt geïnstrueerd om een reeks taken snel en efficiënt uit te voeren in vergelijking met de verwerkingskracht van een mens (Lee, 2011).
Debugging	Debugging is het verwijderen van fouten uit een serie stappen of programma's (Gorissen, Coetsier, Hölsgens, van den Berg, & Kral, 2017)
Decompositie	Decompositie is het opdelen in kleinere onderdelen. De onderdelen kunnen vervolgens afzonderlijk worden begrepen, opgelost, ontwikkeld en geëvalueerd. Dit maakt complexe problemen gemakkelijker op te lossen, nieuwe situaties beter te begrijpen en grote systemen gemakkelijker te ontwerpen (Csizmadia et al., 2015).
Generalisatie	Generalisatie is geassocieerd met het identificeren van patronen, overeenkomsten en verbindingen en het exploiteren van die functies. Het is een manier om snel nieuwe problemen op te lossen op basis van eerdere oplossingen voor problemen en voort te bouwen op eerdere ervaringen. Algoritmen die sommige specifieke problemen oplossen, kunnen worden aangepast om een hele reeks vergelijkbare problemen op te lossen (Csizmadia et al., 2015).

Reeds uitgevoerd onderzoek toont aan dat, wanneer vaardigheden die vallen onder computational thinking verder worden ontwikkeld, het mensen in staat kan stellen om grote hoeveelheden gegevens te analyseren en systemen te verfijnen. En tevens door het verwerken en representeren van gegevens door deze te filteren en gegevens te ontdekken die verstopt of verspreid zijn over de data (Wing, 2008).

Wanneer computational thinking skills worden aangeleerd zal dit resulteren in het vergroten van de probleemoplossende vaardigheden (Yadav, Zhou, Mayfield, Hambruch, & Korb, 2011). Computational thinking moet dan ook worden beschouwd als een fundamentele vaardigheid en daarmee als belangrijk onderdeel van het ontwikkelen van het analytisch denkvermogen van elk kind, evenals het leren lezen, schrijven en rekenen (Wing, 2006).

Vraagstelling en hypotheses

Het bestuderen van de literatuur heeft geleid tot de volgende vraagstelling: “Wat is de invloed van visuele of tastbare output van de programmeeromgeving op de ontwikkeling van computational thinking skills bij leerlingen primair onderwijs in de leeftijd van 9 tot en met 13 jaar en welk effect is daarbij meetbaar op algoritmische vaardigheden, debugging en probleem decompositie??”

De subvragen voor huidig onderzoek zijn als volgt:

- Is er een significant verschil in ontwikkeling van computational thinking skills bij leerlingen meetbaar onder invloed van programmeren?
- Is er een significant verschil meetbaar in de ontwikkeling van de karakteristieken algoritmische vaardigheden, debugging en probleem decompositie wanneer leerlingen programmeren binnen een programmeeromgeving met een visuele of tastbare output?
- Is er een verschil in oplossingsvaardigheid bij een wiskundig, algoritmisch probleem meetbaar wanneer leerlingen programmeren in een programmeeromgeving met een visuele of tastbare output?

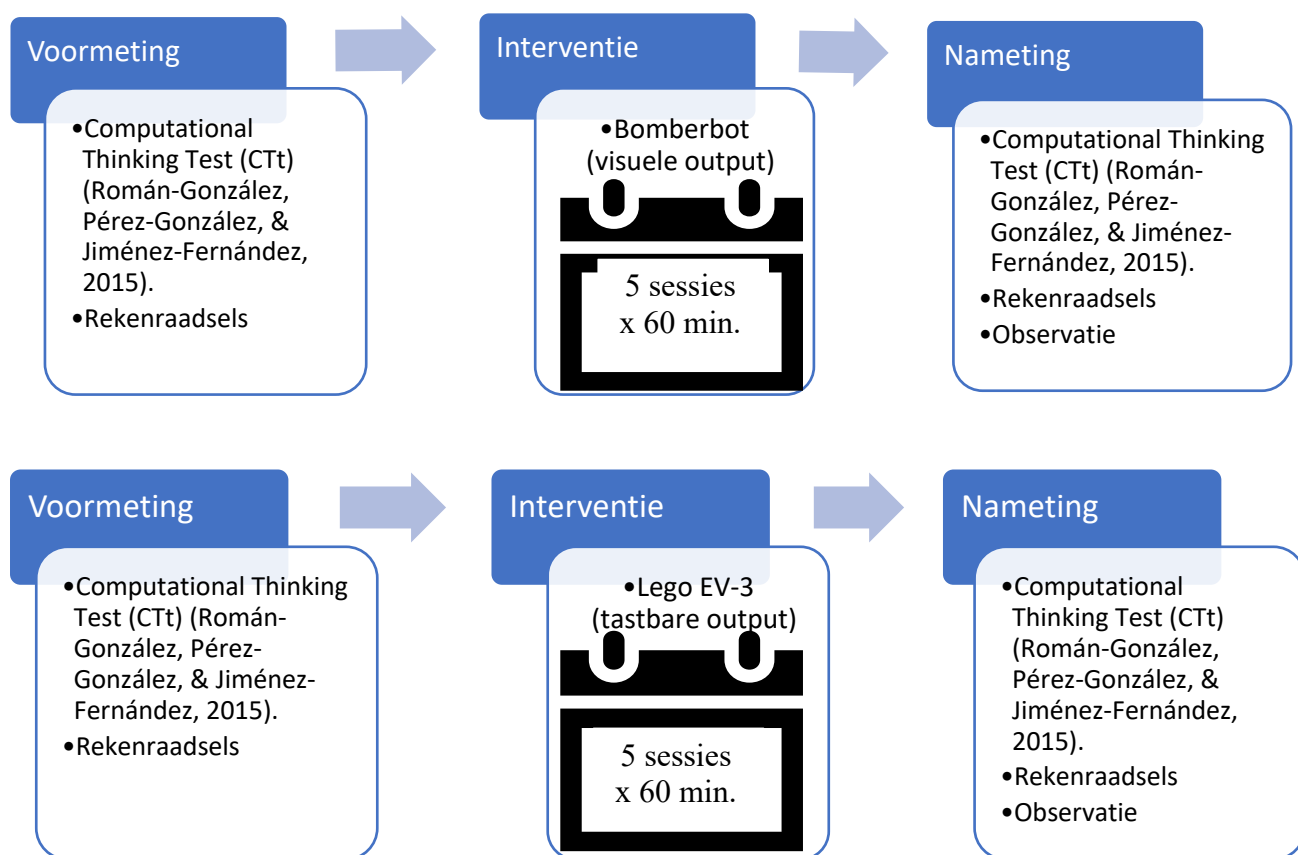
De hypotheses voor huidig onderzoek zijn als volgt:

- Er is een positief verschil in de ontwikkeling van computational thinking skills onder invloed van programmeren.
- De ontwikkeling van algoritmische vaardigheden, debugging en probleemdecompositie ligt in een programmeeromgeving met tastbare output significant hoger dan in een programmeeromgeving met visuele output.
- Programmeren in een programmeeromgeving met tastbare output leidt tot een hogere mate van algoritmische oplossingsvaardigheid dan programmeren in een programmeeromgeving met visuele output.

Methode

Ontwerp

Voor dit onderzoek is gebruik gemaakt van Educational Design Research EDR (Creswell, 2014). Met het onderzoek is onderzocht wat het effect is van het leren programmeren met programmeeromgevingen met verschillende output (visueel/ tastbaar) op de ontwikkeling van de computational thinking skills bij leerlingen uit het primair onderwijs. Om deze ontwikkeling vast te stellen heeft er in dit onderzoek een kwantitatieve voor- en nameting plaatsgevonden op basis van observatie en vragenlijstonderzoek in combinatie met het vinden van een oplossing voor een wiskundig probleem. Door een voormeting en een nameting uit te voeren kon worden bekeken of de interventie effect heeft gehad en of er verschillen zijn in de ontwikkeling van de computational thinking skills tussen het werken met een programmeeromgeving met visuele of tastbare output. De interventie bestond uit vijf keer een uur werken met de, at random toegewezen, programmeeromgeving, op basis van een challenge die voor beide groepen voor zover mogelijk gelijk is gemaakt. De interventie heeft plaatsgevonden over een tijdsbestek van zes weken waarna de nameting heeft plaatsgevonden. Figuur 2 geeft het ontwerp van het onderzoek schematisch weer.



Figuur 2 Schematische weergave onderzoeksontwerp

Participanten

Dit onderzoek is uitgevoerd op een school voor primair onderwijs in het midden van het land met leerlingen van groep 6, 7 en 8 ($N = 52$). De leerlingen binnen deze doelgroep hebben een leeftijd van 9 tot en met 13 jaar ($M = 10.42$, $SD = .98$). De onderzoeksgroep bestaat uit 15 leerlingen uit groep 6, 24 leerlingen uit groep 7 en 13 leerlingen uit groep 8. De leerlingen zijn at random verdeeld over twee onderzoeksgroepen, 1) programmeeromgeving met visuele output ($n = 24$) en 2) programmeeromgeving met tastbare output ($n = 28$). Twee leerlingen, in de betreffende leeftijdscategorie, zijn uitgesloten van onderzoek in verband met het ontbreken van toestemming van ouders. Deze leerlingen zaten beide in de visuele output groep.

Materialen

Interventie

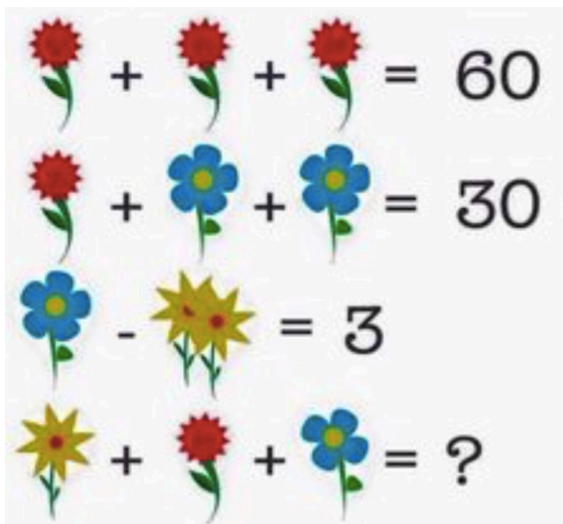
De onafhankelijke variabele in dit onderzoek is de interventie waarin de leerlingen, tijdens een vijftal bijeenkomsten met een tijdsbestek van 60 minuten, werken met een programmeeromgeving met visuele of tastbare output. De programmeeromgeving met visuele output is aangeboden middels het programma Bomberbot. Bomberbot is een online educatief platform waarmee leerkrachten programmeerlessen kunnen verzorgen aan leerlingen in groep 5 t/m 8. Binnen dit programma leren leerlingen fundamentele programmeerconcepten en vaardigheden die zij toepassen in een online game (Bomberbot introductiegids). Het programma Bomberbot kan puur en alleen via beeldschermprogrammeren gebruikt worden. Er is geen fysiek tastbare omgeving aan verbonden. De programmeeromgeving met tastbare output is aangeboden middels de omgeving Lego EV-3. Lego EV-3 is een fysiek aanwezige, programmeerbare PLC voorzien van actuatoren en sensoren waarmee een robot kan worden gebouwd. Een PLC is een elektronisch apparaat in de vorm van een digitale computer. In deze computer is een microprocessor aanwezig die op basis van informatie, die hij via zijn ingangen ontvangt, een aantal uitgangen aanstuurt. Het educatieve Lego-programmeerinterface biedt leerlingen de mogelijkheid om op een eenvoudige manier, binnen een grafische programmeeromgeving, een programmeertaal te leren waardoor leerlingen hun vaardigheid kunnen ontwikkelen in programmeertechnieken (Zimmerman, 2017).

De programmeerbijeenkomsten waren opgebouwd in moeilijkheidsgraad en begonnen allen met een korte introductie van de opdracht. Daarnaast richtte de eerste bijeenkomst zich op de basisprincipes van het programmeren. De vijfde bijeenkomst bestond uit een afsluitende challenge, opgedeeld in vijf deel-challenges die eveneens zijn opgebouwd in moeilijkheidsgraad. Het doel van deze lessen en afsluitende challenge is om door middel van programmeren binnen de gestelde tijd van het beginpunt naar het eindpunt te komen in een te doorlopen 'doolhof'. Binnen het tijdsbestek van een deel-challenge hebben de leerlingen een drietal pogingen om tot de oplossing van de deel-challenge te komen.

Meetinstrumenten

Voor dit onderzoek zijn drie afhankelijke variabelen onderzocht, namelijk: algoritmisch denken, debugging en probleemdecompositie. Voor het onderzoeken van bovenstaande afhankelijke variabelen is gebruik gemaakt van het volgende meetinstrument: Computational Thinking test (CTt) (Román-González et al., 2015). Om de ontwikkeling van algoritmisch denken onder invloed van de programmeer-interventie meer diepgaand te meten, hebben de leerlingen als voor- en nameting diverse rekenraadsels opgelost. De rekenraadsels vragen om een algoritmische oplossing en waren opklimmend in moeilijkheidsgraad (figuur 3). Daarnaast is er tijdens de laatste challenge geobserveerd in hoeverre de leerlingen de geleerde programmeervaardigheden efficiënt en binnen de gestelde tijd hebben opgelost.

Tevens zijn de volgende algemene vragen gesteld: respondentnummer, leeftijd en geslacht van de leerling. Het respondentnummer is gebruikt om de gegevens van de leerlingen van de voor- en nameting aan elkaar te kunnen koppelen.



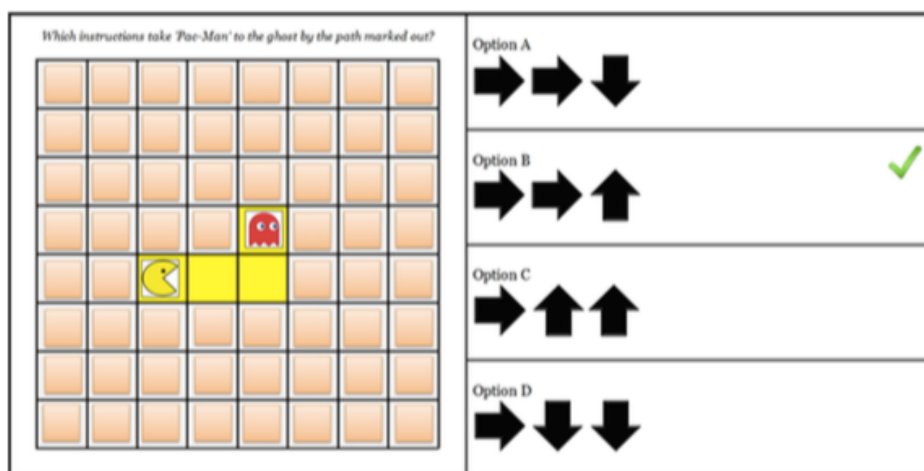
Figuur 3. Voorbeeld rekenraadsel

Computational Thinking test (CTt)

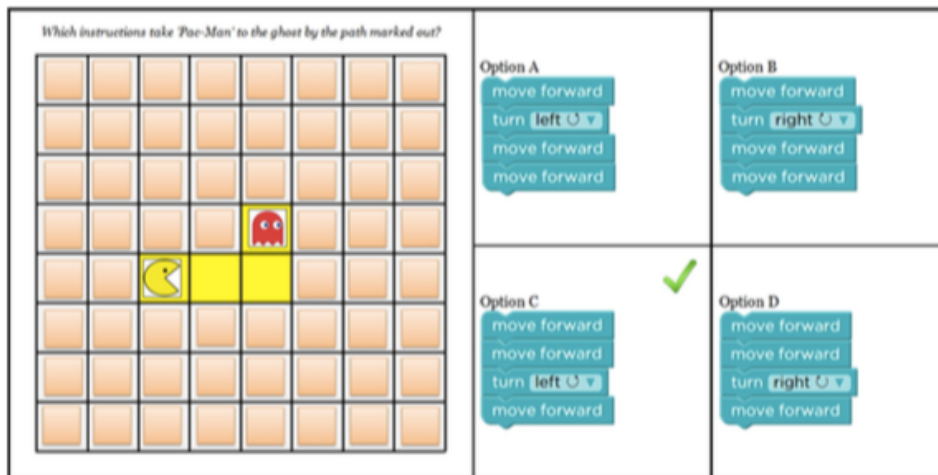
De Computational Thinking test (CTt) (Román-González et. al., 2015) is een meerkeuzetest samengesteld uit 28 items, die ingevuld dienen te worden in een maximale tijd van 45 minuten. Deze test heeft als doel het ontwikkelingsniveau van computational thinking skills te meten. De CTt is in eerste instantie voornamelijk ontworpen voor studenten van 12 tot en met 14 jaar oud (eerste en tweede jaar secundair onderwijs), maar het kan ook worden gebruikt in lagere klassen (groep 5 tot groep 8) (Román-González, Pérez-González, & Jiménez-Fernández, 2017).

Elk item van de CTt wordt gepresenteerd in een 'doolhof' (23 items) of in een 'canvas' (5 items) -interface. Bij elk item worden de antwoorden gepresenteerd door middel van visuele pijlen (figuur 4.1) (8 items) of visuele blokken (figuur 4.2) (20 items). De items zijn verdeeld in de volgende onderdelen: 1) Sequencing: de leerling moet opeenvolgend een reeks van commando's rangschikken (14 items). 2) Voltooing: hierbij dient de leerling een onvolledig gegeven reeks opdrachten in te vullen (9 items). 3) Debugging: de leerling moet fouten opsporen in een onjuist gegeven reeks opdrachten (5 items). De CTt is opbouwend in moeilijkheidsgraad waar de CTt-score kan variëren van 0 tot 28 punten welke quasi-normaal is verdeeld (Román-González et. al., 2017). De betrouwbaarheid van de CTt vragenlijst is hoog ($\alpha = .793$) (Nunnally & Bernstein, 1994; Román-González et. al., 2017).

Voor de CTt vragen is geen duidelijk overzicht welke vraag een beroep doet op welke vaardigheid. Daarom is voor dit onderzoek door een tweetal beoordelaars per vraag gekeken welke vaardigheid wordt getoetst. Hierbij is onderscheid gemaakt tussen: algoritmische vaardigheden, de vaardigheid debugging en de vaardigheid probleemdecompositie. Om de interbeoordelaarsbetrouwbaarheid te controleren, de mate van overeenstemming per vraag tussen het oordeel van de twee personen, is de Cohen's Kappa (Cohens's κ) berekend. Er was een hoge overeenstemming tussen beide beoordelaars, $\kappa = .898$, $p < .01$. Om acceptabel te zijn dient de waarde van Cohen's Kappa hoger te zijn dan 0.6 (McHugh, 2012).



Figuur 4.1 Voorbeeldvraag meetinstrument CTt visuele pijlen



Figuur 4.2 Voorbeeldvraag meetinstrument CTt visuele blokken

Rekenraadsels

Voor het oplossen van een algoritmisch probleem is in dit onderzoek gebruik gemaakt van een vijftal rekenraadsels (figuur 3). Deze rekenraadsels zijn opklimmend in moeilijkheidsgraad en kunnen door toepassing van het correcte te doorgronden algoritme succesvol worden opgelost. Hiervoor dienen de leerlingen een set van stappen middels wiskundige handelingen uit te voeren om het probleem op te lossen (Csizmadia et al., 2015).

Observatie

Tijdens de laatste challenge is geobserveerd in hoeverre de leerlingen de geleerde programmeervaardigheden hebben toegepast en in hoeverre de leerlingen deze challenge zo efficiënt mogelijk hebben opgelost. In beide programmeeromgevingen is gekeken naar of de challenge binnen de beschikbare tijd is opgelost en naar het efficiënt gebruik maken van de programmeermogelijkheden, waaronder het wel/niet toevoegen van onnodige programmeerblokken. Voor de programmeeromgeving met visuele output is per deelchallenge de efficiëntie gebaseerd op het aantal gebruikte programmeerblokken. Hierbij geeft programma Bomberbot een score, middels sterren, waarbij drie sterren inhoudt dat de leerlingen het meest efficiënt gebruik heeft gemaakt van de programmeermogelijkheden (Bomberbot introductiegids).

Voor de programmeeromgeving met tastbare output is tevens per deelchallenge middels handmatig turven van de hoeveelheid gebruikte programmeerblokken de efficiëntie bepaald.

Procedure

Voorafgaand aan het uitvoeren van het onderzoek is het onderzoek ingediend bij de commissie voor ethische toetsing cETO. Na goedkeuring van de commissie voor ethische toetsing is toestemming gevraagd aan de directeur van de school voor het uitvoeren van het onderzoek. Vervolgens zijn alle ouders/ verzorgers van de leerlingen geïnformeerd over het onderzoek middels een brief (bijlage 1)

waarin het doel en het verloop van het onderzoek is omschreven. Bij deze brief is een toestemmingsformulier bijgevoegd. Tevens is er door de onderzoeker een ouderavond georganiseerd waarbij ouders/verzorgers de mogelijkheid hebben gehad voor het stellen van vragen. Voor dit onderzoek is, in verband met de wettelijke tekenbevoegdheid, gebruik gemaakt van twee versies van toestemmingsformulieren (active consent). Als eerste het formulier voor leerlingen onder de 12 jaar en als tweede het formulier voor leerlingen van 12 jaar en ouder. Leerlingen waarvan het toestemmingsformulier niet retour is gekomen naar de onderzoeker hebben niet deelgenomen aan de voor- en nameting. De leerlingen hebben wel deelgenomen aan de interventie omdat dit deel uitmaakte van het onderwijsprogramma van de school.

Het onderzoek is uitgevoerd conform de normen die de ethische commissie heeft voorgeschreven. Het onderzoek is gestart met een voormeting. Deze voormeting is uitgevoerd in het “leerlab” van de school waar voor iedere leerling een laptop met de geopende vragenlijst klaar stond. Voor de rekenvragen was er voor de leerlingen de mogelijkheid om gebruik te maken van kladpapier. Nadat leerlingen de test hadden afgerond zijn zij teruggegaan naar hun klas. De leerlingen zijn at random verdeeld in groepen van twee en at random toegewezen aan de programmeeromgeving met visuele of tastbare output. Vervolgens is de interventie gestart. De interventie bestond uit vijfmaal een uur programmeren binnen de toegewezen programmeeromgeving onder begeleiding van de onderzoeker. Hierbij waren steeds drie groepjes van twee tegelijk aanwezig bij een interventiemoment in het “leerlab”. Waarbij de aanwezigen allen werkten in dezelfde programmeeromgeving.

De interventie bestond uit het uitleggen van de betreffende leeromgeving en het oplossen van opdrachten welke voor elk groepje van de betreffende leeromgeving gelijk waren. De vijfde challenge bestond uit vijf deelchallenges welke opbouwend in moeilijkheidsgraad waren. Hierbij mochten de leerlingen per deelchallenge maximaal drie pogingen gebruiken en hadden zij per deelchallenge vijftien minuten beschikbaar. Wanneer de deelchallenge niet binnen de gestelde eisen werd behaald startte de leerling met de volgende deelchallenge. De leerlingen werden tijdens de interventiemomenten door middel van de challenge-opdrachten gestimuleerd om probleemoplossend te werk te gaan. De onderzoeker heeft tijdens de interventiemomenten een coachende rol aangenomen.

Na deze vijftal challenge opdrachten is er een nameting uitgevoerd. Deze nameting bestond uit het invullen van een vragenlijst en het hernieuwd oplossen van vijf rekenraadsels als algoritmisch rekenprobleem, welke tevens is uitgevoerd in het “leerlab”. Net als bij de voormeting stonden er laptops klaar waarop de vragenlijst kon worden ingevuld. De vragenlijst was exact dezelfde als de leerlingen hebben ingevuld bij de voormeting alleen zonder de vraag over leeftijd en geslacht. Om het leereffect tegen te gaan, zijn bij de rekenraadsels de getallen aangepast, bij de CTt zijn de voorbeeldvragen in een andere volgorde gesteld en zijn de antwoorden van de meeste meerkeuzevragen gespiegeld. Hierbij is gekozen om de meerkeuze antwoorden van de vragen 3, 7, 11,

16 en 19 niet te spiegelen aangezien bij deze vragen de antwoorden in een specifieke volgorde worden gegeven (figuur 5). Voor de rekenraadsels was er, net als bij de voormeting, de mogelijkheid om gebruik te maken van kladpapier. Nadat leerlingen de test hadden afgerond zijn zij teruggegaan naar hun klas. Naast de nameting is er tijdens het laatste interventiemoment een observatie uitgevoerd, hierbij is gelet op het efficiënt gebruiken van de programmeermogelijkheden en het afronden van de deelchallenge binnen de gestelde tijd.



Figuur 5 voorbeeldvraag meetinstrument CTt antwoordmogelijkheid in specifieke volgorde

Data-analyse

De kwantitatieve data in dit onderzoek zijn geanalyseerd door middel van het softwareprogramma Statistical Package for the Social Sciences (SPSS), versie 24 (IBM SPSS Statistics, n.d.).

Voor de start van het onderzoek hebben alle respondenten een respondentnummer toegewezen gekregen. De data is binnen het programma SPSS verwerkt onder vermelding van het respondentnummer. Binnen het programma SPSS is door middel van t-toetsen het effect van het werken met een programmeeromgeving met visuele of tastbare output op de ontwikkeling van computational thinking skills en de ontwikkeling op algoritmische vaardigheden geanalyseerd. Door middel van t-toetsen is antwoord gegeven op de vraag of er een significant verschil is in computational thinking skills in de nameting in vergelijking met de voormeting. Tevens is door middel van t-toetsen geanalyseerd of er een significant verschil is in ontwikkeling van de computational thinking skills tussen de leerlingen die met een programmeeromgeving met visuele of tastbare output hebben gewerkt.

Resultaten

Om de resultaten van de voormeting en de nameting met elkaar te vergelijken zijn er diverse t-toetsen uitgevoerd. De resultaten van de voormeting en de nameting van de CTt van de totale groep, de groep die heeft gewerkt met de programmeeromgeving met visuele output en met tastbare output zijn weergegeven in tabel 2.

Tabel 2

Verschillen tussen de voor- en nameting op de CTt voor de totale, visuele en tastbare groep

Variabelen	n	Voormeting		Nameting		t	p	95% CI		Cohen's d
		M	SD	M	SD			LL	UL	
Totaal CT										
Groep totaal	52	.41	.13	.43	.02	1.29	.20	-.06	.01	.18
Groep visueel	24	.41	.15	.44	.13	.80	.43	-.12	.07	-.16
Groep tastbaar	28	.41	.12	.43	.14	-1.10	.28	-.08	.09	-.21
Debugging										
Groep totaal	52	.41	.21	.42	.18	-.42	.67	-.06	.04	-.06
Groep visueel	24	.38	.21	.43	.16	-1.16	.26	-.14	.04	-.24
Groep tastbaar	28	.44	.20	.42	.19	.83	.42	-.03	.08	.16
Algoritmische v.										
Groep totaal	52	.41	.13	.44	.14	-1.29	.20	-.06	.01	-.18
Groep visueel	24	.41	.16	.44	.13	-.66	.52	-.12	.08	-.13
Groep tastbaar	28	.41	.12	.44	.14	-1.28	.21	-.07	.02	-.24
Probleemdec.										
Groep totaal	52	.32	.23	.33	.19	-.31	.76	-.07	.05	-.04
Groep visueel	24	.32	.24	.35	.19	-.43	.67	-.12	.08	-.09
Groep tastbaar	28	.32	.23	.32	.20	.00	1.00	-.08	.08	.00

Noot. CT = Computational Thinking Skills CI = confidence interval; LL = lower limit; UL = upper limit.

Computational thinking skills

Door middel van een afhankelijke t-toets is geanalyseerd of er een effect is van de interventie op de computational thinking skills bij de nameting in vergelijking met de voormeting bij de totale onderzoeksgroep. Tevens is er gekeken of er een effect is van de interventie op de computational thinking skills bij de groep met de programmeeromgeving met visuele output en de onderzoeksgroep met de programmeeromgeving met tastbare output. Op basis van de vergelijking van de gemiddelden van de voormeting ($M = .41$, $SD = .13$) en nameting ($M = .43$, $SD = .02$) bij de totale onderzoeksgroep

wordt er een positief verschil waargenomen. Echter, na het uitvoeren van de t-test blijkt dit geen significant meetbaar effect, $t(50) = -1.29, p = .20, 95\% \text{ CI } [-.06, .01], d = -.18$. Op basis van de vergelijking van de gemiddelden van de voormeting van de leerlingen die hebben gewerkt met de programmeeromgeving met visuele output ($M = .41, SD = .15$) en nameting ($M = .44, SD = .13$) wordt er een effect waargenomen. Echter, na het uitvoeren van de t-test blijkt dit geen significant meetbaar effect te zijn, $t(23) = .80, p = .43, 95\% \text{ CI } [-.10, .04], d = -.16$. Gebaseerd op de vergelijking van de gemiddelden van de voormeting van leerlingen die hebben gewerkt met de programmeeromgeving met tastbare output ($M = .41, SD = .12$) en nameting ($M = .43, SD = .14$) wordt er een verschil waargenomen. Echter, blijkt na het uitvoeren van de t-test dit geen significant meetbaar verschil, $t(27) = -1.10, p = .28, 95\% \text{ CI } [-.06, .02], d = -.21$.

Om na te gaan of er een effect is op de ontwikkeling van computational thinking skills op de nameting in vergelijking tot de voormeting van leerlingen die hebben gewerkt met de programmeeromgeving met visuele output of leerlingen die hebben gewerkt met de programmeeromgeving met tastbare output zijn beide groepen met elkaar vergeleken middels een onafhankelijke t-toets. De analyse van de t-toets wijst uit dat de leerlingen die hebben gewerkt met de programmeeromgeving met visuele output of tastbare output niet significant meetbaar van elkaar verschillen op de computational thinking skills, $t(50) = .13, p = .90, 95\% \text{ CI } [-.07, .08], d = .04$.

Debugging

Door middel van een afhankelijke t-toets is geanalyseerd of er een effect is van de interventie op de ontwikkeling van de vaardigheid debugging op de nameting ten opzichte van de voormeting van leerlingen die hebben gewerkt met de programmeeromgeving met visuele output of leerlingen die hebben gewerkt met de programmeeromgeving met tastbare output zijn beide groepen met elkaar vergeleken middels een onafhankelijke t-toets. Op basis van de vergelijking van de gemiddelden van de voormeting ($M = .41, SD = .21$) en nameting ($M = .42, SD = .18$) bij de totale onderzoeksgroep wordt er een positief effect waargenomen. Echter, na het uitvoeren van de t-test blijkt dit geen significant meetbaar effect, $t(51) = -.42, p = .67, 95\% \text{ CI } [-.06, .04], d = -.06$. Op basis van de vergelijking van de gemiddelden van de voormeting van de leerlingen die hebben gewerkt met de programmeeromgeving met visuele output ($M = .38, SD = .21$) en nameting ($M = .43, SD = .16$) wordt er een positief effect waargenomen. Na het uitvoeren van de t-test blijkt dit echter geen significant meetbaar effect te zijn, $t(23) = -1.16, p = .26, 95\% \text{ CI } [-.14, .04], d = -.24$. Op basis van de vergelijking van de gemiddelden van de voormeting van leerlingen die hebben gewerkt met de programmeeromgeving met tastbare output ($M = .44, SD = .20$) en nameting ($M = .42, SD = .19$) wordt er een negatief effect waargenomen. Echter, na het uitvoeren van de t-test blijkt ook dit geen significant meetbaar effect, $t(27) = .83, p = .42, 95\% \text{ CI } [-.03, .08], d = .16$.

Om te onderzoeken of er een significant effect is op de ontwikkeling van de vaardigheid debugging op de nameting in vergelijking tot de voormeting van leerlingen die hebben gewerkt met de programmeeromgeving met visuele output of tastbare output zijn beide groepen met elkaar vergeleken door middel van een onafhankelijke t-toets. Uit de analyse van de t-toets blijkt er geen significant verschil is tussen de ontwikkeling van leerlingen die hebben gewerkt met de programmeeromgeving met visuele output en tastbare output op de vaardigheid debugging, $t(50) = 1.42, p = .16, 95\% \text{ CI } [-.03, .17], d = .40$.

Algoritmische vaardigheden

Aan de hand van een afhankelijke t-toets is geanalyseerd of er een effect is van de interventie op de ontwikkeling van de algoritmische vaardigheden op de nameting ten opzichte van de voormeting bij de totale onderzoeksgroep en bij de visuele en tastbare output groep. Op basis van de vergelijking van de gemiddelden van de voormeting ($M = .41, SD = .13$) en nameting ($M = .44, SD = .14$) bij de totale onderzoeksgroep wordt een positief effect waargenomen. Echter, na het uitvoeren van de t-test blijkt dit geen significant meetbaar effect, $t(51) = -1.29, p = .20, 95\% \text{ CI } [-.06, .01], d = -.18$. Gebaseerd op de vergelijking van de gemiddelden van de voormeting ($M = .41, SD = .16$) en nameting ($M = .44, SD = .13$) van de leerlingen die hebben gewerkt met de programmeeromgeving met visuele output wordt een positief effect waargenomen. Na het uitvoeren van de t-test blijkt dit echter geen significant meetbaar effect te zijn, $t(23) = -.66, p = .52, 95\% \text{ CI } [-.09, .05], d = -.13$. Op basis van de vergelijking van de gemiddelden van de voormeting ($M = .41, SD = .12$) en nameting ($M = .44, SD = .14$) van leerlingen die hebben gewerkt met de programmeeromgeving met tastbare output wordt een positief effect waargenomen. Na het uitvoeren van de t-test blijkt ook dit geen significant meetbaar effect, $t(27) = -1.28, p = .21, 95\% \text{ CI } [-.07, .02], d = -.24$.

Om te bepalen of er een significant effect is op de ontwikkeling van de algoritmische vaardigheden op de nameting in vergelijking tot de voormeting van leerlingen die hebben gewerkt met de programmeeromgeving met visuele output of leerlingen die hebben gewerkt met de programmeeromgeving met tastbare output zijn beide groepen met elkaar vergeleken middels een onafhankelijke t-toets. De analyse van de t-toets wijst uit dat de leerlingen die hebben gewerkt met de programmeeromgeving met visuele output of tastbare output niet significant meetbaar van elkaar verschillen op de algoritmische vaardigheden, $t(50) = .11, p = .91, 95\% \text{ CI } [-.08, .07], d = .03$.

Probleemdecompositie

Door middel van een afhankelijke t-toets is geanalyseerd of er een effect is van de interventie op de vaardigheid probleemdecompositie op de nameting ten opzichte van de voormeting bij de totale onderzoeksgroep en bij de onderzoeksgroepen die hebben gewerkt met de programmeeromgeving met visuele output of tastbare output. Op basis van de vergelijking van de gemiddelden van de voormeting

($M = .32$, $SD = .23$) en nameting ($M = .33$, $SD = .19$) bij de totale onderzoeksgroep wordt een positief effect waargenomen. Echter, na het uitvoeren van de t-test blijkt dit geen meetbaar significant effect, $t(51) = -.31$, $p = .76$, 95% CI $[-.07, .05]$, $d = -.04$. Op basis van de vergelijking van de gemiddelden van de voormeting ($M = .32$, $SD = .24$) en nameting ($M = .35$, $SD = .19$) van de leerlingen die hebben gewerkt met de programmeeromgeving met visuele output wordt er een positief effect waargenomen. Echter, na het uitvoeren van de t-test blijkt ook geen significant meetbaar effect, $t(23) = -.43$, $p = .67$, 95% CI $[-.12, .08]$, $d = -.09$. Op basis van de vergelijking van de gemiddelden van de voormeting ($M = .32$, $SD = .23$) en nameting ($M = .32$, $SD = .20$) van leerlingen die hebben gewerkt met de programmeeromgeving met tastbare output wordt geen effect waargenomen. Wederom blijkt na het uitvoeren van de t-test geen significant meetbaar effect, $t(27) = .00$, $p = 1.00$, 95% CI $[-.08, .08]$, $d = .00$.

Om na te gaan of er een significant effect is op de ontwikkeling van de vaardigheid probleemdecompositie op de nameting in vergelijking tot de voormeting van leerlingen die hebben gewerkt met de programmeeromgeving met visuele output of leerlingen die hebben gewerkt met de programmeeromgeving met tastbare output zijn beide groepen met elkaar vergeleken middels een onafhankelijke t-toets. De analyse van de t-toets wijst uit dat er geen significant verschil is tussen de ontwikkeling van leerlingen die hebben gewerkt met de programmeeromgeving met visuele output of leerlingen die hebben gewerkt met de programmeeromgeving met tastbare output op de vaardigheid probleemdecompositie, $t(50) = .33$, $p = .74$, 95% CI $[-.11, .15]$, $d = .09$.

Rekenraadsels

Door middel van een afhankelijke t-toets is geanalyseerd of er een effect is van de interventie op het aantal opgeloste rekenraadsels op de nameting ten opzichte van de voormeting bij de totale onderzoeksgroep en bij de onderzoeksgroepen die hebben gewerkt met de programmeeromgeving met visuele output of tastbare output. De gegevens voor het correct oplossen van de rekenraadsels zijn weergegeven in tabel 3. Op basis van de vergelijking van de gemiddelden van de voormeting bij ($M = .23$, $SD = .19$) en nameting ($M = .24$, $SD = .17$) bij de totale onderzoeksgroep wordt er een positief effect waargenomen. Echter, wordt dit niet ondersteund door de t-toets, $t(50) = -.26$, $p = .80$, 95% CI $[-.07, .05]$, $d = -.04$.

Op basis van de vergelijking van de gemiddelden van de voormeting van de leerlingen die hebben gewerkt met de programmeeromgeving met visuele output ($M = .22$, $SD = .17$) en nameting ($M = .24$, $SD = .19$) van de leerlingen die hebben gewerkt met de programmeeromgeving met visuele output wordt een positief effect waargenomen. Echter, na het uitvoeren van de t-test blijkt er geen significant meetbaar effect, $t(23) = .55$, $p = .59$, 95% CI $[-.12, .07]$, $d = -.11$. Op basis van de vergelijking van de gemiddelden van de voormeting ($M = .24$, $SD = .21$) en nameting ($M = .24$, $SD =$

.16) van leerlingen die hebben gewerkt met de programmeeromgeving met tastbare output wordt geen effect waargenomen. Dit wordt onderkend na het uitvoeren van de t-toets, $t(27) = .18$, $p = .86$, 95% CI $[-.08, .09]$, $d = .03$.

Tabel 3.

Rekenraadsels correct opgelost

		Voormeting								Nameting							
		Rekenraadsel correct opgelost								Rekenraadsel correct opgelost							
Groep	<i>n</i>	<i>M</i>	Totaal	0	1	2	3	4	5	M	Totaal	0	1	2	3	4	5
T. groep	52	.23	60	14	21	12	5	0	0	.24	62	11	24	13	4	0	0
Visueel	24	.22	25	6	11	6	1	0	0	.24	27	6	9	7	2	0	0
Tastbaar	28	.24	34	8	10	6	4	0	0	.24	31	5	15	6	2	0	0

Noot. T. Groep = totale groep; Visueel = Groep visuele output; Tastbaar = Groep tastbare output. *M* = Gemiddeld aantal opgelost ten opzichte van het totale aantal. Totaal = Cumulatief

Om na te gaan of er een significant verschil is tussen de ontwikkeling van leerlingen die hebben gewerkt met de programmeeromgeving met visuele output of leerlingen die hebben gewerkt met de programmeeromgeving met tastbare output van het aantal opgeloste rekenraadsels zijn beide groepen met elkaar vergeleken middels een onafhankelijke t-toets. De analyse van de t-toets wijst uit dat er geen significant verschil is tussen de ontwikkeling van leerlingen die hebben gewerkt met de programmeeromgeving met visuele output of leerlingen die hebben gewerkt met de programmeeromgeving met tastbare output op het aantal correct opgeloste rekenraadsels, $t(50) = .53$, $p = .60$, 95% CI $[-.09, .15]$, $d = .15$.

Observatie challenges (sessie 5)

Tijd

Door middel van een onafhankelijke t-toets is geanalyseerd of er een significant verschil is tussen de programmeeromgevingen, met visuele of tastbare output, in het behalen van de challenges binnen de gestelde tijd. De gegevens voor het oplossen van de deelchallenge binnen de gestelde tijd weergegeven in tabel 4. Op basis van de vergelijking van de gemiddelden blijkt dat de groep die heeft gewerkt met de programmeeromgeving met de visuele output ($M = 5.00$, $SD = .00$) in vergelijking met de groep die heeft gewerkt met de programmeeromgeving met de tastbare output ($M = 4.93$, $SD = .27$) meer challenges binnen de gestelde tijd heeft opgelost. Uit de analyse van de t-toets, voor de behaalde challenges binnen de gestelde tijd, blijkt er geen significant verschil tussen groepen met de

programmeeromgeving met visuele of tastbare output, $t(25) = 1.00$, $p = .34$, 95% CI $[-.08, .23]$, $d = .38$.

Tabel 4.

Deelchallenges binnen de tijd en efficiënt opgelost correct

Groep	<i>n</i>	Deelchallenge binnen de tijd opgelost							Deelchallenge efficiënt opgelost						
		<i>M</i>	<i>SD</i>	1	2	3	4	5	<i>M</i>	<i>SD</i>	1	2	3	4	5
Visuele output	13	5.00	.00	13	13	13	13	13	.60	.24	13	8	5	7	10
Tastbare output	14	4.93	.27	14	13	14	14	14	.96	.09	15	6	2	0	0

Noot. *M* = Gemiddeld aantal opgelost ten opzichte van het totale aantal. *SD* = Standaarddeviatie

Efficiëntie

Door middel van een onafhankelijke t-toets is geanalyseerd of er een significant verschil is tussen de programmeeromgevingen, met visuele of tastbare output, in het gebruik van de meest efficiënte oplossing van de challenges. De gegevens voor het meest efficiënt oplossen van de deelchallenge weergegeven in tabel 4. Op basis van de vergelijking van de gemiddelden blijkt dat de groep die heeft gewerkt met de programmeeromgeving met de visuele output ($M = .60$, $SD = .24$) in vergelijking met de groep die heeft gewerkt met de programmeeromgeving met de tastbare output ($M = 0.96$, $SD = .09$) een hogere score laat zien op efficiëntie. Uit de analyse van de t-toets, voor het gebruik van de meest efficiënte oplossing van de challenges, blijkt er een significant meetbaar verschil, $t(25) = -4.99$, $p = <.01$, 95% CI $[-.51, -.20]$, $d = 1.95$.

Conclusie en discussie

Dit onderzoek had als doel een antwoord te vinden op de vraag wat de invloed is van het verschil tussen visuele en tastbare output van de programmeeromgeving op de ontwikkeling van computational thinking skills bij leerlingen primair onderwijs in de leeftijd van 9 tot en met 13 jaar. De computational thinking skills zijn in huidig onderzoek geoperationaliseerd naar de vaardigheden debugging, algoritmische vaardigheden en probleemdecompositie. Daarnaast is tevens onderzocht of er een meetbaar verschil is in ontwikkeling van computational thinking skills onder invloed van programmeren.

Om antwoord te vinden op de onderzoeksvragen is een drietal hypothesen geformuleerd en onderzocht.

De eerste hypothese luidde: Er is een positief verschil in de ontwikkeling van computational thinking skills onder invloed van programmeren. Deze hypothese is getoetst middels t-toetsen.

Vanuit de gemiddelde scores van de voor en nameting lijkt er een positief effect te zijn. Dit geldt zowel de totale score als voor de onderdelen debugging, algoritmische vaardigheden, probleemdecompositie apart. Echter, na het uitvoeren van de t-toetsen blijkt dit effect niet significant, waardoor deze hypothese toch wordt verworpen.

De tweede hypothese luidde: De ontwikkeling van algoritmische vaardigheden, debugging en probleemdecompositie ligt in een programmeeromgeving met tastbare output significant hoger dan in een programmeeromgeving met visuele output. Deze hypothese is getoetst middels t-toetsen. Een vergelijking van de scores op de voormeting met de scores op de nameting laat een positief effect zien voor algoritmisch denken bij beide programmeeromgevingen. Daarnaast is er voor debugging en probleemdecompositie een positief verschil zichtbaar bij de programmeeromgeving met visuele output. Een bestudering van de means maakt dit ook zichtbaar. Ondanks deze gemeten positieve effecten toont een t-toets analyse aan dat de verschillen die ontstaan op basis van de interventie, hoewel dus meetbaar aanwezig, niet significant zijn waardoor deze statistisch moeten worden verworpen.

De derde en tevens laatste hypothese luidde: Programmeren in een programmeeromgeving met tastbare output leidt tot een hogere mate van algoritmische oplossingsvaardigheid dan programmeren in een programmeeromgeving met visuele output. Vanuit de vergelijking van de meanscores van de nameting ten opzichte van de voormeting is er een positief verschil bij de leerlingen die hebben gewerkt met de programmeeromgeving met visuele output. Uit de resultaten van de t-toetsen blijkt er ondanks het gemeten positieve effect tevens geen significant verschil op algoritmische oplossingsvaardigheden tussen de programmeeromgeving met visuele of tastbare output, hierdoor wordt deze hypothese verworpen.

Het leren programmeren in een programmeeromgeving met visuele of tastbare output heeft ondanks positief meetbare effecten niet het verwachte significante effect gehad op de ontwikkeling van computational thinking skills bij leerlingen primair onderwijs in de leeftijd van 9 tot en met 13 jaar. De interventie heeft geen significant meetbare verschillen opgeleverd op de vaardigheden debugging, probleemdecompositie en de algoritmische vaardigheden. De resultaten uit dit onderzoek komen overeen met de bevindingen van Gorissen en collega's (2017) namelijk dat er na interventie, middels het werken met een programmeeromgeving, geen significante groei kan worden waargenomen op de vaardigheden debugging en probleemdecompositie. Echter is dit in tegenspraak tot het onderzoek van Atmatzidou en Demetriadis (2016) waarin wordt gesteld dat elk soort programmeeromgeving leidt tot een significante positieve ontwikkeling van de computational thinking skills. Een kanttekening die hierbij beschreven wordt is dat de ontwikkeling van computational thinking skills een aanzienlijk aantal trainingssessies vereist, waarbij aanzienlijk niet verder wordt gedefinieerd (Atmatzidou & Demetriadis, 2016; Clark, Tanner-Smith & Killingsworth, 2015).

Mogelijkerwijs is dit de reden dat het aantal trainingssessies (interventies) in het door ons uitgevoerde onderzoek niet voldoende groot is voor een significante ontwikkeling van de computational thinking skills.

Uit de resultaten van dit onderzoek blijkt echter wel een significant verschil tussen de mate van het schrijven van een efficiënt programma. Dit is gebleken op basis van een analyse door de onderzoeker van de door leerlingen geconstrueerde programma's die middels een onafhankelijke t-toets zijn geanalyseerd. De mate van efficiëntie bij het schrijven van een programma in de programmeeromgeving met tastbare output ligt hoger dan in de programmeeromgeving met visuele output. Daaruit kan geconcludeerd worden dat het schrijven van een programma in een programmeeromgeving met tastbare output leidt tot een hogere mate van efficiëntie.

Beperkingen

Het onderzoek kent een aantal beperkingen waardoor de bevindingen met enige voorzichtigheid dienen te worden geïnterpreteerd. In de eerste plaats is huidig onderzoek uitgevoerd bij een kleine groep leerlingen uit het primair onderwijs waardoor dit onderzoek een te lage statistische power had. Om te voldoen aan de power-eisen dienden er voor dit onderzoek 111 participanten mee te doen. Dit heeft mogelijkerwijs effect op gehad op de mate van significantie van de resultaten.

In de tweede plaats zijn deze leerlingen allen van één school afkomstig. Het uitgevoerde onderzoek is daardoor beperkt generaliseerbaar naar leerlingen van andere scholen of andere regio's. Om dit onderzoek breder generaliseerbaar te maken zal dit onderzoek dan ook herhaald moeten worden met respondenten afkomstig van meerdere scholen uit verschillende regio's uit het primair onderwijs.

In de derde plaats is bij huidig onderzoek gekozen voor de Computational Thinking test (CTt) (Román-González et al., 2015) die in 2017 is beoordeeld als valide meetinstrument (Román-González et al., 2017). Deze beoordeling is gebaseerd op het feit dat er op dit moment geen andere valide en toepasbare meetinstrumenten beschikbaar zijn waarmee computational thinking skills op een gevalideerde manier kunnen worden gemeten bij participanten in deze leeftijdscategorie. De CTt kent wel een aantal beperkingen. De CTt meet namelijk alleen het oplossen van computational thinking middels de projectie van logische en visuele problemen. Een voorbeeld hiervan is het oplossen van doolhoven of het tekenen van geometrische patronen (Román-González et al., 2017). De CTt meet hierdoor niet de computational thinking skills in brede zin met een afdekking van alle CT-karakteristieken. Hiervoor zou de CTt zich ook moeten richten op problemen met verschillende functies, waaronder de modellering van wetenschappelijke simulaties, algoritmische samenstellingen van computermuziek en digitale interactieve vertelkunst (Burke, 2012; Edwards, 2011; Howland & Good, 2015; Weintrop et al., 2016). Ondanks de beperkingen van de CTt is de CTt wel het meest geschikte instrument om de computational thinking skills te meten bij leerlingen in de

leeftijdscategorie van huidig onderzoek. Andere beschikbare meetinstrumenten richten zich voornamelijk op leerlingen in hogere leeftijdscategorie of op één specifieke vaardigheid van de computational thinking skills (Voogt, et al., 2017).

Aanbevelingen

Voor dit onderzoek is gekozen voor een interventie van vijf keer 60 minuten. Mogelijk zou het verhogen van het aantal sessies en het verlengen van de tijd een groter en tevens significant verschil kunnen bevorderen tussen het werken met de programmeeromgeving met visuele output en programmeeromgeving met tastbare output. Het is dan ook wenselijk om dit onderzoek te herhalen met meerdere en langere sessies over een langere periode. Dit wordt ondersteund door de resultaten uit het onderzoek van Atmatzidou en Demetriadis (2016) en Skulmowski et al. (2016) waarbij is gekozen voor 11 interventiesessies van twee uur per week.

Tevens is ervoor gekozen om dit onderzoek uit te voeren op één en dezelfde school voor primair onderwijs. Door in vervolgonderzoek een grotere en bredere, meer verspreide populatie te gebruiken wordt de generaliseerbaarheid vergroot. Op die manier kan er voor een bredere populatie geconcludeerd kan worden of er een significant verschil is tussen de groep die werkt met de programmeeromgeving met visuele output en de groep die werkt met de programmeeromgeving met tastbare output op de computational thinking skills debugging, probleemdecompositie en algoritmische vaardigheden. Wanneer er een verschil kan worden aangetoond tussen beide programmeeromgevingen op de ontwikkeling van computational thinking skills kunnen basisscholen een weloverwogen keuze maken om de ontwikkeling van computational thinking skills bij leerlingen te stimuleren.

Uit het uitgevoerde onderzoek blijkt dat programmeren met een programmeeromgeving met visuele of tastbare output geen significant meetbaar effect heeft op de computational thinking skills debugging, probleemdecompositie en algoritmische vaardigheden. Mogelijk heeft deze interventie wel een positief effect op andere karakteristieken van computational thinking, maar deze zijn niet als zodanig specifiek gemeten. Aanbeveling voor vervolgonderzoek is dan ook om het effect op meerdere karakteristieken van computational thinking te onderzoeken. Deze informatie draagt bij aan het gerichter kunnen aanleren van specifieke vaardigheden wat de efficiëntie van het onderwijs ten goede komt. Het verdient dan ook de aanbeveling om dit onderzoek te herhalen echter dan gericht op andere karakteristieken van computational thinking.

Tevens is het waardevol om verder te onderzoeken wat maakt dat de efficiëntie van het geschreven programma binnen de programmeeromgeving met tastbare output hoger ligt in vergelijking met de programmeeromgeving met visuele output. Het schrijven van een efficiënter programma verkleint wellicht de kans op het maken van fouten in het geschreven programma waardoor het makkelijker is om de fouten in het geschreven programma op te sporen en dit

uiteindelijk leidt tot een efficiënter programma Dit kan de motivatie van leerlingen om te programmeren bevorderen waardoor leerlingen mogelijk een grotere ontwikkeling van de computational thinking skills doormaken.

Referenties

- Anderson, R. (2008). Implications of the information and knowledge society for education. In J. Voogt & G. Knezek (Eds.), *International Handbook of Information Technology in Primary and Secondary Education* (pp. 5–22). New York: Springer.
- Ariza, C., Collins, S., Dehlinger, J., Dierbach, C., Hochheiser, H., Jerome, G., ... Kleinasser, W. (2011). A model for piloting pathways for computational thinking in a general education curriculum. *Proceedings of the 42nd ACM technical symposium on Computer science education*, 257-262. Doi:10.1145/1953163.1953243
- Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75, 661-670.
- Barmby, P., Harries, T., Higgins, S., & Suggate, J. (2007). How can we assess mathematical understanding?. In J. H. Woo, H. C. Lew, K. S. Park, & D. Y. Seo (Eds.), *Proceedings of the 31st Conference of the International Group for the Psychology of Mathematics Education* (Vol. 2, pp. 41-48). Seoul, Korea: PME.
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48-54.
- Bassok, M., & Holyoak, K. J. (1989). Interdomain transfer between isomorphic topics in algebra and physics. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 15(1), 153-166.
- Berland, M., & Lee, V. (2011). Collaborative strategic board games as a site for distributed computational thinking. *International Journal of Game-Based Learning*, 1(2), 65-81. doi:10.4018/ijgbl.2011040105
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). Developing computational thinking in compulsory education - Implications for policy and practice. *JRC Science for Policy Report*.
- Bomberbot introductiegids. Verkregen van:
<https://drive.google.com/file/d/0ByRx0jG7Swn0c28yOFdqRDNrUzA/view> op 10 november 2018.
- Brennan, K., & Resnick, M. (2012, April). *New frameworks for studying and assessing the development of computational thinking*. Paper presented at the Annual meeting of the American Educational Research Association, Vancouver, BC, Canada.
- Burke, Q. (2012). The markings of a new pencils: Introducing programming as writing in the middle school classroom. *Journal of Media Literacy Education*, (4)2, 121-135.

- Chalkiadaki, A. (2018). A systematic literature review of 21st century skills and competencies in primary education. *International Journal of Instruction*, 11(3), 1-16.
doi:10.12973/iji.2018.1131a
- Clark, D. B., Tanner-Smith, E. E., & Killingsworth, S. S. (2016). Digital games, design, and learning: A systematic review and meta-analysis. *Review of Educational Research*, 86(1), 79-122. doi:10.3102/0034654315582065
- Clements D.H., & Gullo, D.F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational Psychology*, 76(6), 1051-1058. doi:10.1037/0022-0663.76.6.1051
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). Computational thinking-A guide for teachers.
- Creswell, J. W. (2014). Educational research: Planning, conducting and evaluating quantitative and qualitative research. Harlow, Groot-Brittannië: Pearson Education.
- Dierbach, C., Hochheiser, H., Collins, S., Jerome, G., Ariza, C., Kelleher, T., ... & Kaza, S. (2011, maart). A model for piloting pathways for computational thinking in a general education curriculum. *Proceedings of the 42nd ACM technical symposium on Computer science education*, 257-262. doi:10.1145/1953163.1953243
- Edwards, M. (2011). Algorithmic composition: Computational thinking in music. *Communications of the ACM*, 54(7), 58-67. doi:10.1145/1965724.1965742
- European Council (2006). Recommendation of the European Parliament and the Council of 18 December 2006 on key competencies for lifelong learning. *Brussels: Official Journal of the European Union*, 30(12), 2006.
- Goldstone, R. L., & Son, J. Y. (2005). The transfer of scientific principles using concrete and idealized simulations. *The Journal of the learning sciences*, 14(1), 69-110.
- Gordon, J., Halász, G., Krawczyk, M., Leney, T., Michel, A., Pepper, D., ... & Wiśniewski, J. (2009). Key competences in Europe: Opening doors for lifelong learners across the school curriculum and teacher education. *CASE network Reports*, 87, 328.
- Gorissen, P., Coetsier, N., Hölsgens, R., van den Berg, W., & Kral, M. (2017). *Doorbraakproject computational thinking (CT) en programmeren in het basisonderwijs*. (Onderzoeks Rapport Nr. 405-15-823). Verkregen van de Nationaal Regieorgaan Onderwijsonderzoek website: <https://www.nro.nl/wp-content/uploads/2018/05/405-15-823-010-School-6.5>
Doorbraakproject-Computational-Thinking-CT-en-Programmeren-in-het-basisonderwijs.pdf
- Grover, S., & Pea, R. (2013). Computational thinking in K-12, A review of the state of the field. *Educational Researchers*, 42(1), 38-43. doi:10.3102/0013189X124663051

- Howland, K., & Good, J. (2015). Learning to communicate computationally with flip: A bi-modal programming language for game creation. *Computers & Education*, 80, 224-240.
doi:10.1016/j.compedu.2014.08.014
- Hsin, I., & Paas, F. (2015). Effects of computer-based visual representation on mathematics learning and cognitive load. *Journal of Educational Technology & Society*, 18(4), 70-77.
- Ishii, H., & Ullmer, B. (1997, March). Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems* (pp. 234-241). ACM. doi:10.1145/258549.258715
- Jeuring, J., Corbalan, G., van Es, N., Leeuwestein, H., & van Montfort, J. (2016). *Leren programmeren in het po- een literatuurreview*. Utrecht: Universiteit Utrecht.
- Khandelwal, M., & Mazalek, A. (2007, februari). Teaching Table: A tangible mentor for pre-K math education. In *Proceedings of the 1st international conference on Tangible and embedded interaction* (pp. 191-194). ACM.
- Kelleher, C., Pausch, R., Pausch, R., & Kiesler, S. (2007, april). Storytelling alice motivates middle school girls to learn computer programming. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 1455-1464). ACM. doi:10.1145/1240624.1240844
- Kennisnet (2016). *Computational thinking in het Nederlandse onderwijs*. Zoetermeer: Kennisnet
- Koninklijke Nederlandse Akademie van Wetenschappen (2012). *Digitale geletterdheid in het voortgezet onderwijs. Vaardigheden en attitudes voor de 21^e eeuw*. Amsterdam: KNAW.
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., ... Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2(1), 32–37.
doi:10.1145/1929887.1929902
- Lindberg, R. S. N., Laine, T. H., & Haaranen, L. (2018). Gamifying programming education in K-12: A review of programming curricula in seven countries and programming games: Gamifying programming education in K-12. *British Journal of Educational Technology*, 1-17.
doi:10.1111/bjet.12685
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51-61.
doi:10.1016/j.chb.2014.09.012
- Marshall, P. (2007, February). Do tangible interfaces enhance learning?. In *Proceedings of the 1st international conference on Tangible and embedded interaction* (pp. 163-170). ACM.
- McHugh, M. L. (2012). Interrater reliability: the kappa statistic. *Biochemia Medica*, 22(3), 276-282.
- Nunnally, J. C., & Bernstein, I. H. (1994). *Psychometric theory* (3rd ed.). New York: McGraw-Hill.

- Okita, S. Y. (2014). The relative merits of transparency: Investigating situations that support the use of robotics in developing student learning adaptability across virtual and physical computing platforms. *British Journal of Educational Technology*, 45(5), 844-862.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc..
- Rogers, Y., Scaife, M., Gabrielli, S., Smith, H., & Harris, E. (2002). A conceptual framework for mixed reality environments: designing novel learning activities for young children. *Presence: Teleoperators & Virtual Environments*, 11(6), 677-686. doi:10.1162/105474602321050776
- Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2015, October). Test de pensamiento computacional: diseño y psicometría general. In *III Congreso Internacional sobre Aprendizaje, Innovación y Competitividad (CINAIC 2015)*.
- Román-González, M., Pérez-González, J., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, 72, 678-691. doi:10.1016/j.chb.2016.08.047
- Sapounidis, T., Demetriadis, S., & Stamelos, I. (2015). Evaluating children performance with graphical and tangible robot programming tools. *Personal and Ubiquitous Computing*, 19(1), 225-237. doi:10.1007/s00779-014-0774-3
- Perković, L., Settle, A., Hwang, S., & Jones, J. (2010, June). A framework for computational thinking across the curriculum. *Paper gepresenteerd op de fifteenth annual conference on Innovation and technology in computer science education*, 123-127.
- Shute, J., Sun, C. & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142-158.
- Skulmowski, A., Pradel, S., Kühnert, T., Brunnett, G., & Rey, G. D. (2016). Embodied learning using a tangible user interface: The effects of haptic perception and selective pointing on a spatial learning task. *Computers & Education*, 92-93, 64-75. doi:10.1016/j.compedu.2015.10.011
- SLO (2015). Computational thinking. Verkregen van: <http://curriculumvandetoekomst.slo.nl/21e eeuwse-vaardigheden/digitale geletterdheid/computational-thinking>.
- Suzuki, H., & Kato, H. (1995). Interaction-level support for collaborative learning: AlgoBlock-an open programming language. In *The first international conference on Computer support for collaborative learning* (pp. 349-355). L. Erlbaum Associates Inc..
- Voorwinden, R. (2016). Nieuw model 21^e eeuwse vaardigheden. Verkregen van: <https://www.kennisnet.nl/artikel/nieuw-model-21e-eeuwse-vaardigheden/>.

- Thijs, A. M., Fisser, P. H. G., & van der Hoeven, M. (2014). *21e eeuwse vaardigheden in het curriculum van het funderend onderwijs*. SLO, nationaal expertisecentrum leerplanontwikkeling.
- Trilling, B., & Fadel, C. (2010). *21st century skills: Learning for life in our times*. San Francisco: Jossey-Bass.
- Ullmer, B., & Ishii, H. (2000). Emerging frameworks for tangible user interfaces. *IBM systems journal*, 39(3.4), 915-931.
- UNESCO. (2011). ICT competency framework for teachers: Unesco.
- van Laar, E., van Deursen, A. J. A. M., van Dijk, J. A. G. M., & de Haan, J. (2017). The relation between 21st-century skills and digital skills: A systematic literature review. *Computers in Human Behavior*, 72, 577-588. doi:10.1016/j.chb.2017.03.010
- Voogt, J., Brand-Gruwel, S., & Van Strien, J. (2017). Effecten van programmeeronderwijs op computational thinking: een reviewstudie. Den Haag: NRO.
- Voogt, J., & Roblin, N. P. (2012). A comparative analysis of international frameworks for 21st century competences: Implications for national curriculum policies. *Journal of Curriculum Studies*, 44(3), 299-321. doi:10.1080/00220272.2012.668938
- Voorwinden, R. (2016). Nieuw model 21^e eeuwse vaardigheden. Verkregen op 31-10-2018 van <https://www.kennisnet.nl/artikel/nieuw-model-21e-eeuwse-vaardigheden/>
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1), 127-147. doi:10.1007/s10956-015-9581-5
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717-3725.
- Yadav, A., Zhou, N., Mayfield, C., Hambruch, S., & Korb, J. (2011). Introducing computational thinking in education courses. Paper gepresenteerd op de 42nd ACM technical symposium on Computer science education, 465-470. doi:10.1145/1953163.1953297
- Zimmerman, B. (2017). Programming education for students of any age: LEGO education Mindstorms EV3 robotics. *Journal of Computing Sciences in Colleges*, 33(1), 42-43.
- Zuckerman, O., Arida, S., & Resnick, M. (2005, april). Extending tangible interfaces for education: digital montessori-inspired manipulatives. Paper gepresenteerd op de SIGCHI conference on Human factors in computing systems, 859-868. doi:10.1145/1054972.1055093

Bijlage 1: Informatiebrief ouder(s)/ verzorger(s)

Informatiebrief voor ouders met betrekking tot het onderzoek 'programmeeronderwijs in tweedimensionale of driedimensionale programmeeromgevingen; de effecten op computational thinking'.

Almere, 15 maart 2019

Geachte ouder(s)/ verzorger(s),

Als student van de Open Universiteit (OU) voer ik een onderzoek uit rondom de effecten van programmeeronderwijs op het logisch benaderen en oplossen van problemen met behulp van ICT-technieken en -gereedschappen. Het onderzoek wordt uitgevoerd op de basisschool van uw kind(eren), in groep 6 t/m 8. Door middel van deze brief wil ik u informeren over het doel en de inhoud van het onderzoek.

In dit onderzoek zullen leerlingen robots gaan programmeren of programmeren op de computer. Waarna wordt onderzocht of het programmeren de rekenvaardigheid verbetert, wat het effect is op het oplossen van problemen en het opdelen van problemen in kleinere deeltappen. Het wordt zeer op prijs gesteld als u uw medewerking wilt verlenen aan dit onderzoek.

Wat wordt er gevraagd?

Wij vragen toestemming voor de deelname van uw kind voor het invullen van de vragenlijsten. Uw kind zal de digitale vragenlijsten invullen vooraf en na het werken met de programmeeromgeving. Dit zal per keer ongeveer 1 uur in beslag nemen. De vragen hebben betrekking op de rekenvaardigheden en de probleemoplossingsvaardigheden van uw kind. Zo vragen we bijvoorbeeld welke stappen een poppetje moet volgen om in een doolhof van punt A naar punt B te komen en om rekenraadsels op te lossen. Vervolgens krijgt uw kind 4 keer een uur programmeeronderwijs per week in één van de twee beschikbare omgevingen met daarnaast een afsluitende challenge op het gebied van programmeren welke tevens een uur zal duren. Na de vijf keer te hebben gewerkt in de programmeeromgeving zullen wij uw kind vragen de vragenlijst nogmaals in te vullen. Deze vijf uur vallen onder het reguliere onderwijsprogramma tijdens schooltijd. Ook het invullen van de vragenlijsten zal plaatsvinden onder schooltijd.

Uw kind zal samen met een andere leerling werken in de programmeeromgeving. De vragenlijst zal beide keren individueel worden ingevuld door uw kind, op een rustige plek in de school. In totaal zal uw kind ongeveer 7 uur bezig zijn verdeeld over een periode van ongeveer 8 weken. In de digitale vragenlijst zal tevens worden gevraagd naar de leeftijd en het geslacht van uw kind. Wanneer het onderzoek is afgelopen zal uw kind ook nog werken in de andere programmeeromgeving, hiermee wordt voorkomen dat kinderen achterstand oplopen ten opzichte van de kinderen die met de andere programmeeromgeving hebben gewerkt.

Hoe doet uw kind mee?

Bijgesloten bij deze brief vindt u het toestemmingsformulier om toestemming te geven voor de deelname van uw kind. Wanneer uw kind 12 jaar of ouder is, dient het kind zelf ook een aparte toestemmingsverklaring te ondertekenen. Het is voor ons van belang dat uw kind meedoet aan het onderzoek.

Vrijwillige deelname en privacy

De deelname van uw kind is vrijwillig en wordt anoniem verwerkt. Wanneer u toestemming verleent kunt u deze toestemming op elk moment intrekken. Aan het begin van het onderzoek wordt aan uw kind gevraagd of hij/zij mee wil doen en hij/zij mag op elk moment aangeven of hij/zij mee wil (blijven) doen of niet. De ervaring leert echter dat kinderen vaak met plezier meedoen aan dergelijke onderzoeken.

Het onderzoek is getoetst door de cETO van de Open Universiteit. Alle informatie wordt vertrouwelijk verwerkt. In de vragenlijsten zal niet worden gevraagd naar de naam van uw kind maar om een respondentnummer, waardoor de gegevens worden losgekoppeld van de naam van uw kind. De gegevens worden in een beveiligde omgeving versleuteld opgeslagen op de server van de Open Universiteit voor de duur van 10 jaar. Daarnaast zijn de gegevens alleen toegankelijk voor de onderzoeker zelf. In onze rapportages praten we altijd over iedereen tegelijkertijd en nooit over u of uw gezin alleen. Voor de privacy verwijs ik u naar de website van de Open Universiteit: <https://www.ou.nl/persoonsgegevens-disclaimer>

Alvast hartelijk bedankt voor uw medewerking!

Heeft u nog vragen, dan kunt u te allen tijde contact opnemen met Michelle Petrus (onderzoek19.3@ou.nl)

Met vriendelijke groet,

Michelle Petrus
(student onderwijswetenschappen, Open Universiteit)

Verantwoordelijke onderzoekers:

Marcus Specht, Msc (begeleiding), Open Universiteit

Nardie Farnchamps, Msc (begeleiding), Open Universiteit

Bijlage 2: Vragenlijst voor en nameting

Voormeting onderzoek Programmeeronderwijs

In deze test ga je verschillende vragen beantwoorden. Je begint eerst met 3 vragen over jezelf. Daarna volgen er vijf rekenraadsels. Na deze rekenraadsels krijg je 28 meerkeuzevragen over programmeren. Succes!
Er zijn 42 vragen in deze enquête.

Algemene vragen

Wat is het nummer dat je van de juf hebt gekregen? *

Vul uw antwoord hier in:

Hoe oud ben je? *

Kies één van de volgende antwoorden

Kies één van de volgende mogelijkheden:

- ☐ 8
- ☐ 9
- ☐ 10
- ☐ 11
- ☐ 12
- ☐ 13

*

Kies één van de volgende mogelijkheden:

- ☐ Vrouw
- ☐ Man

Rekenraadsels voor

Deze test bestaat uit 5 rekenraadsels.

Vanaf het begin van de test heb je 20 minuten om het beste te doen wat je kunt. Het is niet belangrijk dat je alle vragen beantwoordt.

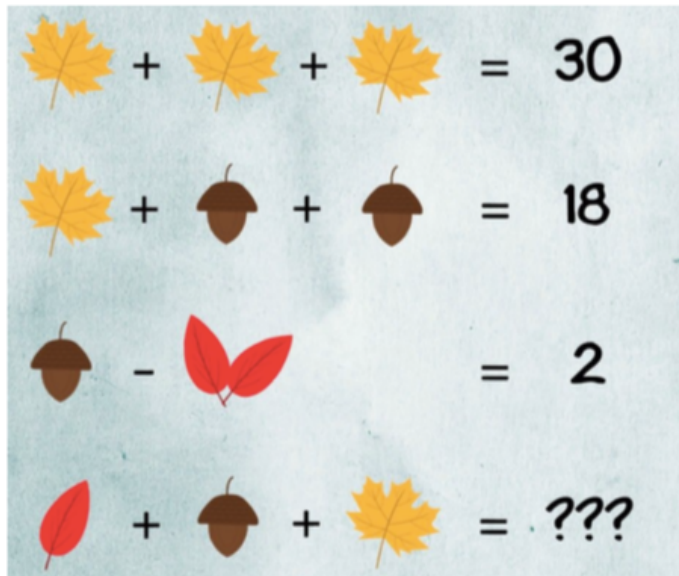
Om door te gaan naar de volgende vraag, klik je onder aan de pagina op de knop 'Volgende'.

HEEL BELANGRIJK: wanneer je klaar bent of wanneer de tijd voorbij is, moet je verder gaan naar de laatste pagina en op de knop 'Verzenden' klikken om je antwoorden op te slaan.

Als je op een vraag moet inzoomen om deze te vergroten, drukt je op 'Ctrl +' op uw toetsenbord (of op 'Ctrl' om het kleiner te maken).

Voordat we de test starten, zien je één voorbeeld, zodat je vertrouwd kunt raken met het soort

raadsels in deze test.
SUCCES!



Een eerste voorbeeldvraag. Beantwoordt deze vraag: *

Vul uw antwoord hier in:

Voorbeeldraadsel

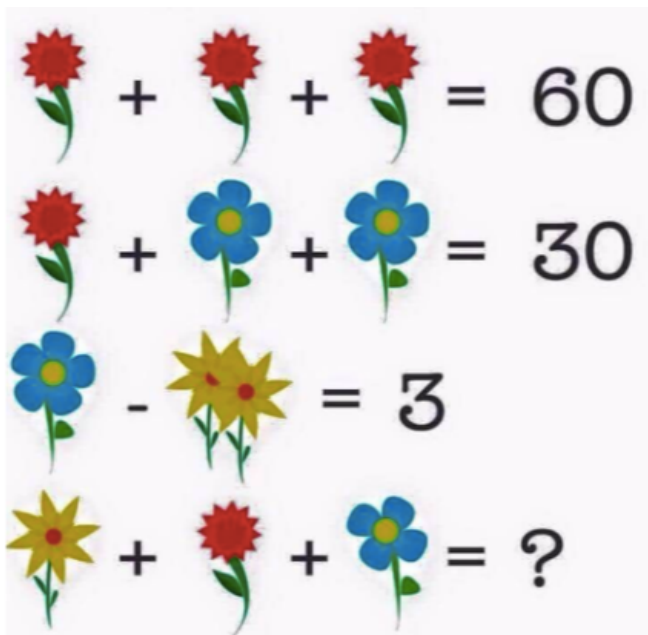
Het goede antwoord van dit raadsel is 15.

Gele blaadje= 10

Eikel= 4

Rode blaadje= 1

Dus $1+4+10= 15$. Je vult hierboven dus het getal 15 in.



Vul uw antwoord hier in:

$$\text{apple} + \text{apple} + \text{apple} = 30$$

$$\text{apple} + \text{banana} + \text{banana} = 18$$

$$\text{banana} - \text{coconut} = 2$$

$$\text{coconut} + \text{apple} + \text{banana} = ?$$

Vul uw antwoord hier in:

$$3 \times \text{cherry} = 27$$

$$3 \times \text{clover} + \text{cherry} = 24$$

$$\text{cherry} + \text{clover} + \text{orange} + \text{orange} = 96$$

$$\text{orange} + \text{cherry} + \text{clover} = ???$$

Vul uw antwoord hier in:

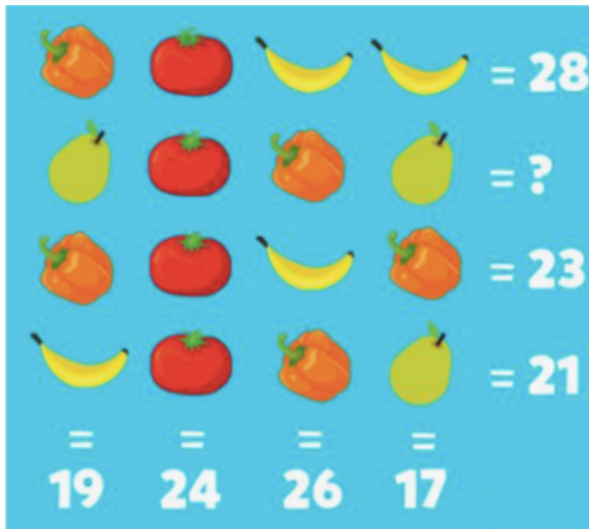
$$\text{plum} + \text{plum} = 6$$

$$\text{orange} - \text{orange} = 2$$

$$2 \times \text{apple} = 10$$

$$\text{plum} \times \text{orange} \times \text{apple} = ?$$

Vul uw antwoord hier in:



Vul uw antwoord hier in:

Computational thinking test

Deze test bestaat uit 28 vragen.

Alle vragen hebben 4 antwoordopties (A, B, C of D) waarvan er slechts één goed is.

Vanaf het begin van de test heb je 45 minuten om het beste te doen wat je kunt. Het is niet belangrijk dat je alle vragen beantwoordt.

Om door te gaan naar de volgende vraag, klik je onder aan de pagina op de knop 'Volgende'.

HEEL BELANGRIJK: wanneer je klaar bent of wanneer de tijd voorbij is, moet je verder gaan naar de laatste pagina en op de knop 'Verzenden' klikken om je antwoorden op te slaan.

Als je op een vraag moet inzoomen om deze te vergroten, druk je op 'Ctrl +' op uw toetsenbord (of op 'Ctrl' om het kleiner te maken).

Voordat we de test starten zie je drie voorbeelden zodat je vertrouwd kunt raken met het soort vragen in deze test en met de personages die zullen verschijnen.

SUCCES!



'Pac-Man'



Geest



Artiest

Welke instructies brengen 'Pac-Man' naar de geest via het gele pad?

Optie A
→ → ↓
Optie B
→ → ↑ ✓
Optie C
→ ↑ ↑
Optie D
→ ↓ ↓

Klik op het goede antwoord (in dit voorbeeld is het goede antwoord B).

*

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

In het eerste voorbeeld word je gevraagd welke instructies 'PacMan' naar de geest brengt op het pad dat is uitgestippeld.

Dat wil zeggen, om 'PacMan' PRECIES te brengen naar het plein waar de geest is (zonder er overheen te gaan of te stoppen) en alleen het gele pad volgen (zonder de wanden, oranje vierkanten, aan te raken).

Het juiste antwoord in dit voorbeeld is B. Klik op de knop "B" onder de vraag.

Welke instructies brengen 'Pac-Man' naar de geest via het gele pad?

Optie A	Optie B
beweeg vooruit	beweeg vooruit
draai links ↺	draai rechts ↻
beweeg vooruit	beweeg vooruit
beweeg vooruit	beweeg vooruit
Optie C ✓	Optie D
beweeg vooruit	beweeg vooruit
beweeg vooruit	beweeg vooruit
draai links ↺	draai rechts ↻
beweeg vooruit	beweeg vooruit

Klik op het goede antwoord (in dit voorbeeld is het goede antwoord C).

*

Kies één van de volgende mogelijkheden:

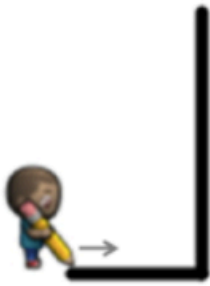
- ☐ A

- ☐ B
- ☐ C
- ☐ D

In het tweede voorbeeld word je opnieuw gevraagd welke instructies 'PacMan' naar de geest brengt door het pad uit te stippelen. In deze vraag worden de opties alleen gepresenteerd als groepen van instructieblokken in plaats van pijlen.

Onthoud dat de vraag is om 'PacMan' PRECIES naar het veld waar de geest is te brengen (zonder eroverheen te gaan of te stoppen) en alleen het gele pad te volgen (zonder de muren te raken, aangeduid door de oranje vierkanten) te volgen.

Het juiste antwoord in dit voorbeeld is C. Klik op de knop "C" onder de vraag.

<p>Welke instructies moet de artiest volgen om de vorm te tekenen? De korte kant is 50 pixels en de lange kant is 100 pixels.</p> 	<p>Optie A</p> <p>✓</p> <p>beweeg vooruit met 50 pixels draai links met 90 graden beweeg vooruit met 100 pixels</p>	<p>Optie B</p> <p>beweeg vooruit met 50 pixels draai rechts met 90 graden beweeg vooruit met 100 pixels</p>
	<p>Optie C</p> <p>beweeg vooruit met 100 pixels draai links met 90 graden beweeg vooruit met 50 pixels</p>	<p>Optie D</p> <p>beweeg vooruit met 100 pixels draai rechts met 90 graden beweeg vooruit met 50 pixels</p>

Klik op het goede antwoord (in dit voorbeeld is het goede antwoord A).

*

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

In het derde voorbeeld word je gevraagd welke instructies de kunstenaar moet volgen om de vorm op het scherm te tekenen. Dat wil zeggen, hoe het potlood te verplaatsen om de vorm te tekenen. De BEWEEG-instructie duwt het potlood tijdens het tekenen, terwijl de SPRING-instructie de artiest springt naar een andere locatie zonder tekenen. De grijze pijl geeft de richting van de eerste beweging

van het potlood aan. De Het juiste antwoord in dit voorbeeld is A. Klik op de knop "A" onder de vraag.

<p>Welke instructies brengen 'Pac-Man' naar de geest via het gele pad?</p>	<p>Optie A</p> <p>→ → ↓</p>
	<p>Optie B</p> <p>→ ↓ ↓</p>
	<p>Optie C</p> <p>→ → ↓ ↓</p>
	<p>Optie D</p> <p>↓ ↓ →</p>

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>Welke stap mist er in de instructies hieronder om 'Pac-Man' naar de geest te brengen via het gele pad?</p> <p>← ← ↑ ¿? → → →</p>	<p>Optie A</p> <p>→</p>
	<p>Optie B</p> <p>←</p>
	<p>Optie C</p> <p>↑</p>
	<p>Optie D</p> <p>↓</p>

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

De instructies moeten 'Pac-Man' naar de geest brengen via het gele pad.
In welke stap van de instructies zit een *fout*?

beweeg vooruit →
 draai links ↶ →
 beweeg vooruit →
 beweeg vooruit →
 draai links ↶ →
 beweeg vooruit →

Stap A
 Stap B
 Stap C
 Stap D

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

Welke instructies moet de artiest volgen om het vierkant te tekenen? Alle kanten van het vierkant zijn 100 pixels lang.

Optie A
 beweeg vooruit met 100 pixels
 draai rechts met 90 graden
 beweeg vooruit met 100 pixels
 draai links met 90 graden
 beweeg vooruit met 100 pixels
 draai rechts met 90 graden
 beweeg vooruit met 100 pixels

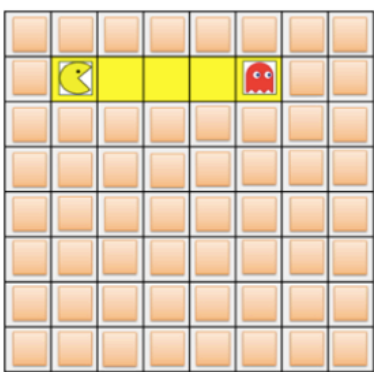




Optie B
 beweeg vooruit met 25 pixels
 draai rechts met 90 graden
 beweeg vooruit met 25 pixels
 draai links met 90 graden
 beweeg vooruit met 25 pixels
 draai rechts met 90 graden
 beweeg vooruit met 25 pixels

Optie C
 beweeg vooruit met 50 pixels
 draai rechts met 90 graden
 beweeg vooruit met 50 pixels
 draai rechts met 90 graden
 beweeg vooruit met 50 pixels
 draai rechts met 90 graden
 beweeg vooruit met 50 pixels

Optie D
 beweeg vooruit met 100 pixels
 draai rechts met 90 graden
 beweeg vooruit met 100 pixels
 draai rechts met 90 graden
 beweeg vooruit met 100 pixels
 draai rechts met 90 graden
 beweeg vooruit met 100 pixels

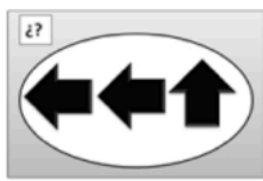
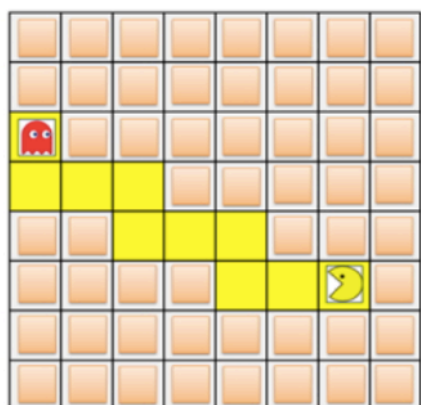
Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>Welke instructies brengen 'Pac-Man' naar de geest via het gele pad?</p> 	<p>Optie A</p> 	<p>Optie B</p> 
	<p>Optie C</p> 	<p>Optie D</p> 

Kies één van de volgende mogelijkheden:


- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>Hoe vaak moet de <u>sequence</u> worden herhaald om 'Pac-Man' naar de geest te brengen via het gele pad?</p>  	<p>Optie A</p> <p>X 2</p>
	<p>Optie B</p> <p>X 1</p>
	<p>Optie C</p> <p>X 4</p>
	<p>Optie D</p> <p>X 3</p>

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

De instructies moeten de artiest de volgende rechthoek 1 keer laten tekenen (50 pixels breed en 100 pixels hoog). In welke stap van de instructies zit een *fout*?



```

herhaal 4 keer
doe
  beweeg vooruit met 50 pixels
  draai links met 90 graden
  beweeg vooruit met 100 pixels
  draai links met 90 graden

```

Stap A is indicated by an arrow pointing to the 'herhaal 4 keer' block.

Stap B is indicated by an arrow pointing to the 'draai links met 90 graden' block after the first 'beweeg vooruit met 50 pixels'.

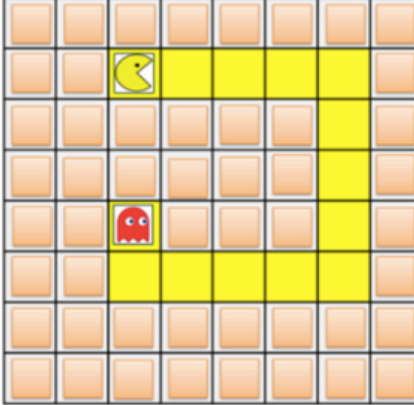
Stap C is indicated by an arrow pointing to the 'beweeg vooruit met 100 pixels' block.

Stap D is indicated by an arrow pointing to the 'draai links met 90 graden' block after the 'beweeg vooruit met 100 pixels'.

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

Welke instructies brengen 'Pac-Man' naar de geest via het gele pad?



Optie A

```

herhaal 4 keer
doe
  herhaal 3 keer
    beweeg vooruit
  draai rechts
beweeg vooruit

```

Optie B

```

herhaal 3 keer
doe
  herhaal 4 keer
    beweeg vooruit
  draai rechts
beweeg vooruit

```

Optie C

```

herhaal 3 keer
doe
  herhaal 4 keer
    beweeg vooruit
  draai rechts
beweeg vooruit

```

Optie D

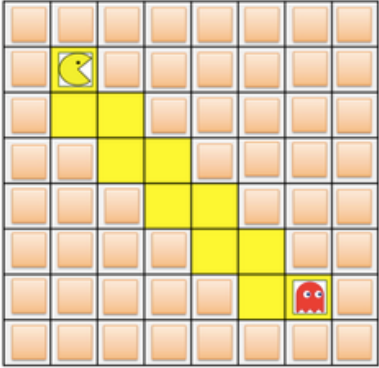








```

herhaal 4 keer
doe
  beweeg vooruit
  herhaal 3 keer
    draai rechts
beweeg vooruit

```

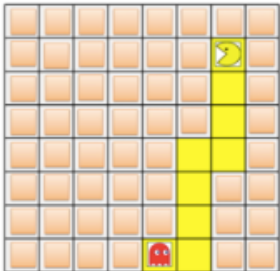
Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>Welke instructies brengen 'Pac-Man' naar de geest via het gele pad?</p> 	<p>Optie A</p> <div data-bbox="651 300 880 427"> <p>Herhaal tot de...</p>   </div>	<p>Optie B</p> <div data-bbox="906 300 1136 427"> <p>Herhaal tot de...</p>   </div>
	<p>Optie C</p> <div data-bbox="651 510 880 638"> <p>Herhaal tot de...</p>   </div>	<p>Optie D</p> <div data-bbox="906 510 1136 638"> <p>Herhaal tot de...</p>   </div>

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>Welke instructie mist om 'Pac-Man' naar de geest te brengen via het gele pad?</p> <div data-bbox="411 1003 580 1218"> <p>Herhaal tot de...</p> <p>doe</p> <p>draai links</p> <p>beweeg vooruit</p> <p>?</p> <p>beweeg vooruit</p> <p>draai rechts</p> <p>beweeg vooruit</p> </div> 	<p>Optie A</p> <div data-bbox="708 1048 900 1099"> <p>draai links</p> </div>	<p>Optie B</p> <div data-bbox="963 1048 1171 1099"> <p>draai rechts</p> </div>
	<p>Optie C</p> <div data-bbox="708 1339 916 1391"> <p>beweeg vooruit</p> </div>	<p>Optie D</p> <p>Er mist geen stap</p>

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

De instructies moeten 'Pac-Man' naar de geest brengen via het gele pad. In welke stap van de instructies zit een **fout**?

Kies één van de volgende mogelijkheden:

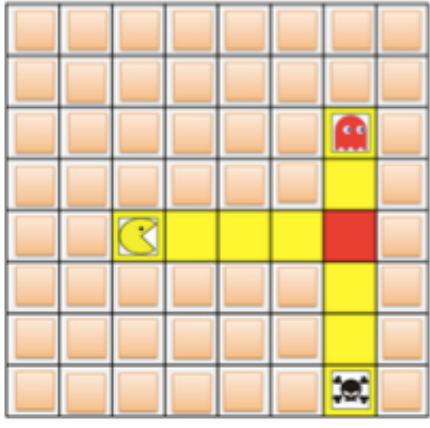
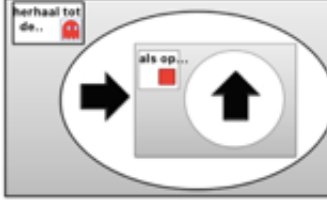
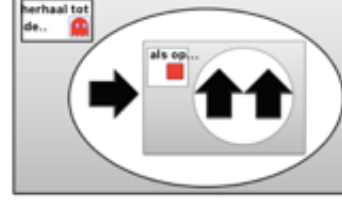
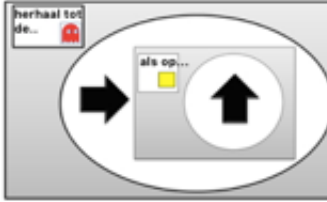
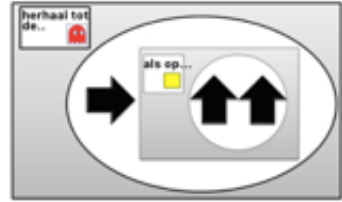
- ☐ A
- ☐ B
- ☐ C
- ☐ D

Welke instructies moet de artiest volgen om de ladder te tekenen die naar de bloem leidt? Er zitten 30 pixels tussen iedere trede.

Optie A	Optie B
<pre> herhaal tot de bloem doe herhaal 4 keer doe beweeg vooruit met 30 pixels draai rechts met 90 graden spring vooruit met 30 pixels </pre>	<pre> herhaal tot de bloem doe herhaal 4 keer doe beweeg vooruit met 120 pixels draai rechts met 90 graden spring vooruit met 30 pixels </pre>
Optie C	Optie D
<pre> herhaal tot de bloem doe herhaal 3 keer doe beweeg vooruit met 30 pixels draai rechts met 90 graden spring vooruit met 210 pixels </pre>	<pre> herhaal tot de bloem doe herhaal 7 keer doe beweeg vooruit met 30 pixels draai rechts met 90 graden spring vooruit met 30 pixels </pre>

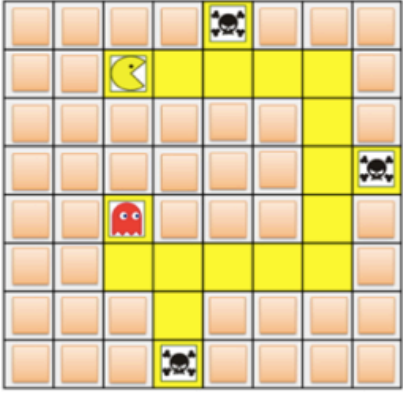




Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>Welke instructies brengen 'Pac-Man' naar de geest via het gele pad?</p> 	<p>Optie A</p> 	<p>Optie B</p> 
	<p>Optie C</p> 	<p>Optie D</p> 

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>Welke instructies brengen 'Pac-Man' naar de geest via het gele pad?</p> 	<p>Optie A</p> 	<p>Optie B</p> 
	<p>Optie C</p> 	<p>Optie D</p> 

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

Wat is de missende instructie om Pas-Man naar de geest te brengen via het gele pad?

Optie A

Optie B

Optie C

Optie D

Zowel optie A en optie C zijn goed

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

De instructies moeten 'Pac-Man' naar de geest brengen via het gele pad. In welke stap van de instructies zit een **fout**?

```

herhaal tot
doe
  beweeg vooruit
  als pad naar links
  doe draai links
  als pad naar rechts
  doe beweeg vooruit
  
```

→ Stap A

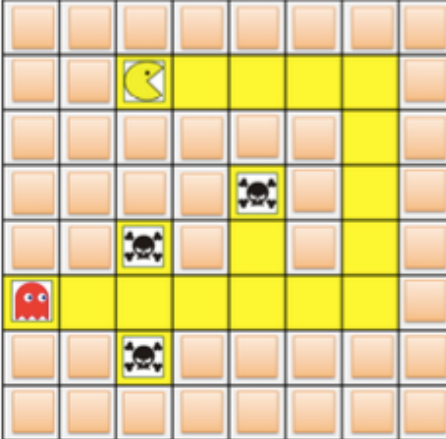
→ Stap B

→ Stap C

→ Stap D

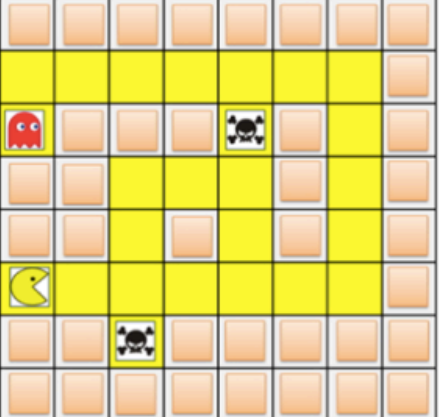
Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>Welke instructies brengen 'Pac-Man' naar de geest via het gele pad?</p> 	<p>Optie A</p> <pre> herhaal tot doe als pad vooruit doe beweeg vooruit anders draai links </pre>	<p>Optie B</p> <pre> herhaal tot doe als pad vooruit doe beweeg vooruit anders draai rechts </pre>
	<p>Optie C</p> <pre> herhaal tot doe als pad naar rechts doe draai rechts anders beweeg vooruit </pre>	<p>Optie D</p> <pre> herhaal tot doe als pad naar links doe draai links anders beweeg vooruit </pre>

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>Welke instructies brengen 'Pac-Man' naar de geest via het gele pad?</p> 	<p>Optie A</p> <pre> herhaal tot doe als pad vooruit doe beweeg vooruit anders draai links </pre>	<p>Optie B</p> <pre> herhaal tot doe als pad vooruit doe beweeg vooruit anders draai rechts </pre>
	<p>Optie C</p> <pre> herhaal tot doe als pad naar rechts doe draai rechts anders beweeg vooruit </pre>	<p>Optie D</p> <pre> herhaal tot doe als pad naar links doe draai links anders beweeg vooruit </pre>

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

De instructies moeten 'Pac-Man' naar de geest brengen via het gele pad. In welke stap van de instructies zit een *fout*?

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

Welke instructie mist om 'Pac-Man' naar de geest te brengen via het gele pad?

Optie A	Optie B
beweeg vooruit	draai rechts
Optie C	Optie D
draai links	Er mist geen stap

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>Welke instructies brengen 'Pac-Man' naar de aardbeien via het gele pad en vertel 'Pac-Man' om alle getoonde aardbeien te eten?</p>	<p>Optie A</p> <pre> als pad vooruit doe herhaal 5 keer doe beweeg vooruit herhaal 3 keer doe Eet 1 aardbei </pre>	<p>Optie B</p> <pre> als pad vooruit doe beweeg vooruit herhaal 3 keer doe Eet 1 aardbei </pre>
	<p>Optie C</p> <pre> als pad vooruit doe herhaal 3 keer doe beweeg vooruit herhaal 5 keer doe Eet 1 aardbei </pre>	<p>Optie D</p> <pre> als pad vooruit doe beweeg vooruit herhaal 3 keer doe Eet 1 aardbei </pre>

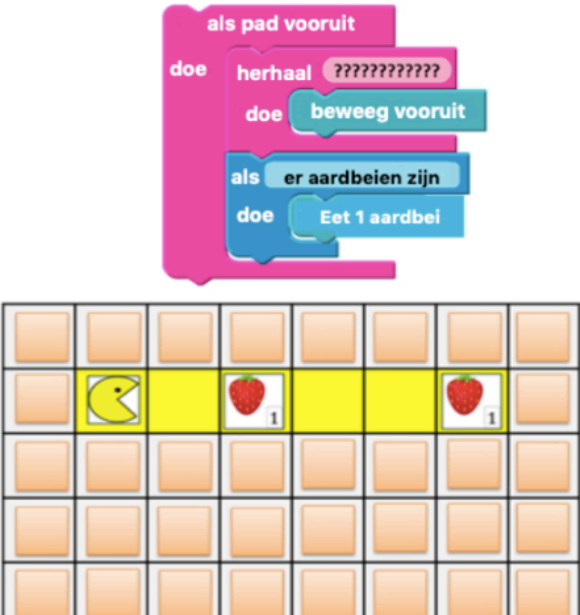
Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>Welke instructies brengen 'Pac-Man' naar de aardbeien via het gele pad en vertel 'Pac-Man' om alle getoonde aardbeien te eten?</p>	<p>Optie A</p> <pre> als pad vooruit doe herhaal 5 keer doe beweeg vooruit herhaal 3 keer doe Eet 1 aardbei </pre>	<p>Optie B</p> <pre> als pad vooruit doe beweeg vooruit herhaal 3 keer doe Eet 1 aardbei </pre>
	<p>Optie C</p> <pre> als pad vooruit doe herhaal 3 keer doe beweeg vooruit herhaal 5 keer doe Eet 1 aardbei </pre>	<p>Optie D</p> <pre> als pad vooruit doe beweeg vooruit herhaal 3 keer doe Eet 1 aardbei </pre>

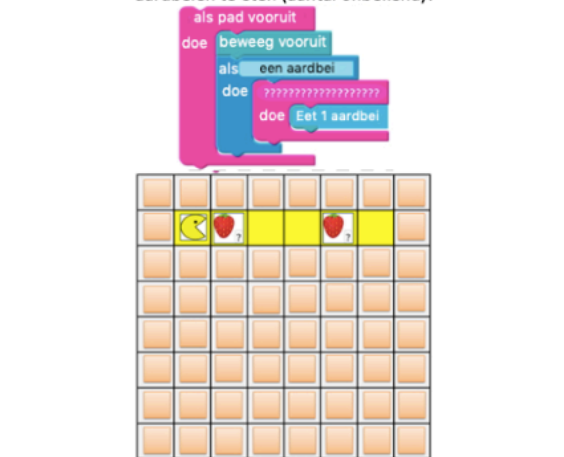
Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>Wat mist er in de instructies hieronder om 'Pac-Man' naar de aardbeien te brengen via het gele pad en vertel 'Pac-Man' om alle getoonde aardbeien te eten?</p>  <p>The script consists of a pink 'als pad vooruit' block followed by a 'doe' block containing a 'herhaal' block with 7 'doe' blocks: 'beweeg vooruit', 'als er aardbeien zijn', and 'doe Eet 1 aardbei'. The grid is an 8x8 orange field with a yellow path starting at (2,2) and ending at (7,7). Strawberries are at (3,3) and (7,7), both labeled with a '1'.</p>	<p>Optie A</p> <p>1 keer</p> <p>Optie B</p> <p>2 keer</p> <p>Optie C</p> <p>3 keer</p> <p>Optie D</p> <p>5 keer</p>
--	---

Kies één van de volgende mogelijkheden:

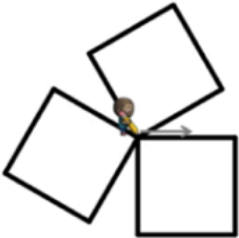
- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>Wat mist er in de instructies hieronder om 'Pac-Man' naar de aardbeien te brengen via het gele pad en vertel 'Pac-Man' om alle getoonde aardbeien te eten (aantal onbekend)?</p>  <p>The script consists of a pink 'als pad vooruit' block followed by a 'doe' block containing 'beweeg vooruit', 'als een aardbei', and 'doe Eet 1 aardbei'. The grid is a 10x10 orange field with a yellow path starting at (2,2) and ending at (5,5). Strawberries are at (2,3) and (5,4), both labeled with a '7'.</p>	<p>Optie A</p> <p>wanneer pad vooruit</p> <p>Optie B</p> <p>wanneer geen pad vooruit</p> <p>Optie C</p> <p>wanneer aardbeien</p> <p>Optie D</p> <p>Wanneer geen aardbeien</p>
--	---

Kies één van de volgende mogelijkheden:


- ☐ A
- ☐ B

- ☐ C
- ☐ D

<p>De volgende de set van instructies welke 'mijn functie' worden genoemd, tekent een vierkant van 100 pixels aan elke kant:</p> <pre> functie mijn functie herhaal 4 keer doe beweeg vooruit met 100 pixels draai rechts met 90 graden </pre> <p>Welke instructies zou de artiest moeten volgen om het volgende ontwerp te tekenen?</p> <p>Elke kant van elk vierkant is 100 pixels lang.</p> 	<p>Optie A</p> <pre> herhaal 3 keer doe mijn functie draai rechts met 120 graden </pre>	<p>Optie B</p> <pre> herhaal 3 keer doe mijn functie draai rechts met 120 graden </pre>
	<p>Optie C</p> <pre> herhaal 4 keer doe mijn functie draai rechts met 90 graden </pre>	<p>Optie D</p> <pre> herhaal 3 keer doe mijn functie draai rechts met 90 graden </pre>

Kies één van de volgende mogelijkheden:

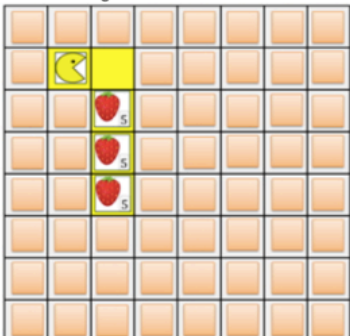
- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>De volgende de set van instructies welke 'mijn functie' worden genoemd, tekent een driehoek van 50 pixels aan elke kant:</p> <pre> Functie mijn functie herhaal 3 keer doe beweeg vooruit met 50 pixels draai links met 120 graden </pre> <p>De onderstaande instructies moeten ervoor zorgen dat de artiest het volgende ontwerp tekent. Elke kant van de driehoek is 50 pixels lang. Wat is de missende instructie?</p> <pre> herhaal ??? keer doe mijn functie spring vooruit met 50 pixels </pre> 	<p>Optie A</p> <p>15</p>	<p>Optie B</p> <p>5</p>
	<p>Optie C</p> <p>4</p>	<p>Optie D</p> <p>3</p>

Kies één van de volgende mogelijkheden:

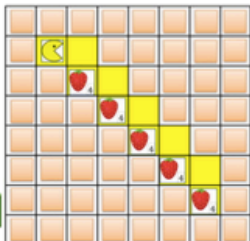
- ☐ A
- ☐ B
- ☐ C

- D

<p>De volgende set van instructies wordt 'krijg 5' genoemd:</p> <pre> Functie krijg 5 herhaal 5 keer doe Eet 1 aardbei </pre> <p>Welke instructies brengen 'Pac-Man' naar de aardbeien via het gele pad en vertellen 'Pac-Man' om alle getoonde aardbeien te eten?</p> 	<p>Optie A</p> <pre> beweeg vooruit draai rechts herhaal 3 keer doe beweeg vooruit krijg 5 </pre>	<p>Optie B</p> <pre> beweeg vooruit draai rechts herhaal 3 keer doe krijg 5 beweeg vooruit </pre>
	<p>Optie C</p> <pre> beweeg vooruit draai rechts herhaal 5 keer doe beweeg vooruit krijg 5 </pre>	<p>Optie D</p> <pre> beweeg vooruit draai rechts herhaal 5 keer doe krijg 5 beweeg vooruit </pre>

Kies één van de volgende mogelijkheden:

- A
- B
- C
- D

<p>De volgende set van instructie wordt 'beweeg en krijg 4' genoemd:</p> <pre> functie beweeg en krijg 4 beweeg vooruit draai rechts beweeg vooruit herhaal 4 keer doe Eet 1 aardbei draai links </pre> <p>Wat is de missende instructie hieronder om 'Pac-Man' naar de aardbeien via het gele pad te brengen en 'Pac-Man' te vertellen om alle getoonde aardbeien te eten?</p>  <pre> herhaal ??? keer doe beweeg en krijg 4 </pre>	<p>Optie A</p> <p>3</p>	<p>Optie B</p> <p>4</p>
	<p>Optie C</p> <p>5</p>	<p>Optie D</p> <p>6</p>

Kies één van de volgende mogelijkheden:

- A
- B
- C
- D

Je bent klaar. Je kan nu weer naar de klas.

Verzend uw enquête.

Bedankt voor uw deelname aan deze enquête.

Nameting onderzoek programmeeronderwijs

Je gaat nu de vragenlijst nog een keer invullen. Ook nu krijg je weer vijf rekenraadsels en 28 programmeervragen. Succes!
Er zijn 40 vragen in deze enquête.

Algemene vragen

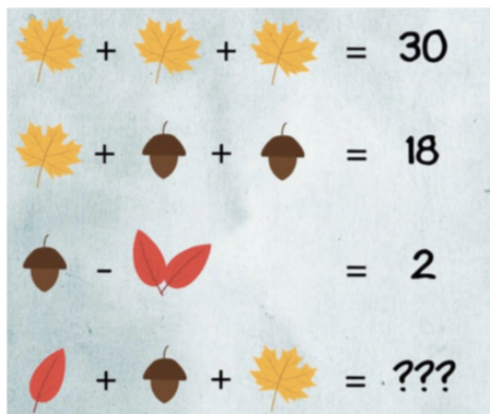
Welk nummer heb je van de juf gekregen? *

Vul uw antwoord hier in:

Deze test bestaat uit 28 vragen. Alle vragen hebben 4 antwoordopties (A, B, C of D) waarvan er slechts één goed is. Vanaf het begin van de test heb je 45 minuten om het beste te doen wat je kunt. Het is niet belangrijk dat je alle vragen beantwoordt. Om door te gaan naar de volgende vraag, klik je onder aan de pagina op de knop 'Volgende'. **HEEL BELANGRIJK:** wanneer je klaar bent of wanneer de tijd voorbij is, moet je verder gaan naar de laatste pagina en op de knop 'Verzenden' klikken om je antwoorden op te slaan. Als je op een vraag moet inzoomen om deze te vergroten, drukt je op 'Ctrl +' op uw toetsenbord (of op 'Ctrl' om het kleiner te maken). Voordat we de test starten zie je drie voorbeelden zodat je vertrouwd kunt raken met het soort vragen in deze test en met de personages die zullen verschijnen. **SUCCES!**

Rekenraadsels

Deze test bestaat uit 5 rekenraadsels. Vanaf het begin van de test heb je 20 minuten om het beste te doen wat je kunt. Het is niet belangrijk dat je alle vragen beantwoordt. Om door te gaan naar de volgende vraag, klik je onder aan de pagina op de knop 'Doorgaan'. **HEEL BELANGRIJK:** wanneer je klaar bent of wanneer de tijd voorbij is, moet je verder gaan naar de laatste pagina en op de knop 'Verzenden' klikken om je antwoorden op te slaan. Als je op een vraag moet inzoomen om deze te vergroten, druk je op 'Ctrl +' op uw toetsenbord (of op 'Ctrl' om het kleiner te maken). Voordat we de test starten, zien je één voorbeeld, zodat je vertrouwd kunt raken met het soort raadsels in deze test. **SUCCES!**



Vul uw antwoord hier in:

Het goede antwoord van dit raadsel is 15. Oranje blaadje= 1 Eikel= 4 Gele blaadje= 10. Dus 1+4+10= 15. Je vult hieronder dus het getal 15 in.

$$\begin{array}{rcl}
 \text{Red Flower} + \text{Red Flower} + \text{Red Flower} & = & 90 \\
 \text{Red Flower} + \text{Blue Flower} + \text{Blue Flower} & = & 40 \\
 \text{Blue Flower} - \text{Yellow Flower} & = & 3 \\
 \text{Yellow Flower} + \text{Red Flower} + \text{Blue Flower} & = & ?
 \end{array}$$

Vul uw antwoord hier in:

$$\begin{array}{rcl}
 \text{Apple} + \text{Apple} + \text{Apple} & = & 60 \\
 \text{Apple} + \text{Banana} + \text{Banana} & = & 28 \\
 \text{Banana} - \text{Coconut} & = & 2 \\
 \text{Coconut} + \text{Apple} + \text{Banana} & = & ?
 \end{array}$$

















Vul uw antwoord hier in:

$$\begin{array}{rcl}
 \text{Cherry} + \text{Cherry} + \text{Cherry} & = & 24 \\
 \text{Clover} + \text{Clover} + \text{Clover} + \text{Cherry} & = & 23 \\
 \text{Cherry} + \text{Clover} + \text{Orange} + \text{Orange} & = & 75 \\
 \text{Orange} + \text{Cherry} + \text{Clover} & = & ???
 \end{array}$$

Vul uw antwoord hier in:

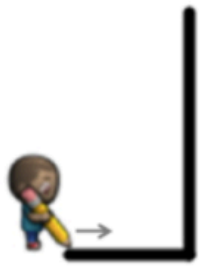
$$\begin{array}{rcl}
 \text{Plum} & + & \text{Half Plum} = 9 \\
 \text{Orange} & - & \text{Half Orange} = 3 \\
 \text{Two Apples} & + & \text{Half Apple} = 15 \\
 \text{Plum} \times \text{Orange} \times \text{Apple} & = & ?
 \end{array}$$

Vul uw antwoord hier in:

				= 28
				= ?
				= 25
				= 23
= 21	= 28	= 26	= 19	

Vul uw antwoord hier in:

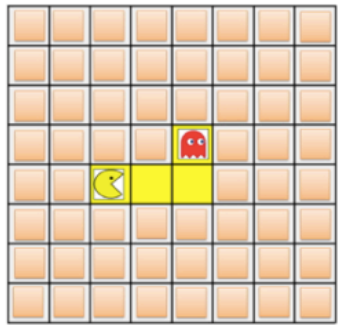
Programmeervragen

<p>Welke instructies moet de artiest volgen om de vorm te tekenen? De korte kant is 50 pixels en de lange kant is 100 pixels.</p> 	<p>Optie A</p> <p>beweeg vooruit met 50 pixels draai links met 90 graden beweeg vooruit met 100 pixels</p>	<p>Optie B</p> <p>beweeg vooruit met 50 pixels draai rechts met 90 graden beweeg vooruit met 100 pixels</p>
	<p>Optie C</p> <p>beweeg vooruit met 100 pixels draai links met 90 graden beweeg vooruit met 50 pixels</p>	<p>Optie D</p> <p>beweeg vooruit met 100 pixels draai rechts met 90 graden beweeg vooruit met 50 pixels</p>

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

In het eerste voorbeeld word je gevraagd welke instructies de kunstenaar moet volgen om de vorm op het scherm te tekenen. Dat wil zeggen, hoe het potlood te verplaatsen om de vorm te tekenen. De BEWEEG-instructie duwt het potlood tijdens het tekenen, terwijl de SPRING-instructie de artiest springt naar een andere locatie zonder tekenen. De grijze pijl geeft de richting van de eerste beweging van het potlood aan. De Het juiste antwoord in dit voorbeeld is A. Klik op de knop "A" onder de vraag.

<p>Welke instructies brengen 'Pac-Man' naar de geest via het gele pad?</p> 	<p>Optie A</p> <p>beweeg vooruit draai links beweeg vooruit beweeg vooruit</p>	<p>Optie B</p> <p>beweeg vooruit draai rechts beweeg vooruit beweeg vooruit</p>
	<p>Optie C</p> <p>beweeg vooruit beweeg vooruit draai links beweeg vooruit</p>	<p>Optie D</p> <p>beweeg vooruit beweeg vooruit draai rechts beweeg vooruit</p>

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

In het tweede voorbeeld word je opnieuw gevraagd welke instructies 'PacMan' naar de geest brengt door het pad uit te stippelen. In deze vraag worden de opties alleen gepresenteerd als groepen van instructieblokken in plaats van pijlen.

Onthoud dat de vraag is om 'PacMan' PRECIES naar het veld waar de geest is te brengen (zonder eroverheen te gaan of te stoppen) en alleen het gele pad te volgen (zonder de muren te raken, aangeduid door de oranje vierkanten) te volgen.
Het juiste antwoord in dit voorbeeld is C. Klik op de knop "C" onder de vraag.

Welke instructies brengen 'Pac-Man' naar de geest via het gele pad?	
	Optie A
	Optie B
	Optie C
	Optie D

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

In het derde voorbeeld word je gevraagd welke instructies 'PacMan' naar de geest brengt op het pad dat is uitgestippeld.

Dat wil zeggen, om 'PacMan' PRECIES te brengen naar het plein waar de geest is (zonder eroverheen te gaan of te stoppen) en alleen het gele pad volgen (zonder de wanden, oranje vierkanten, aan te raken).

Het juiste antwoord in dit voorbeeld is B. Klik op de knop "B" onder de vraag.

Welke instructies brengen 'Pac-Man' naar de geest via het gele pad?	
	Optie A
	Optie B
	Optie C
	Optie D

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>Welke stap mist er in de instructies hieronder om 'Pac-Man' naar de geest te brengen via het gele pad?</p> <div style="text-align: center; margin-bottom: 10px;"> </div>	<div style="margin-bottom: 10px;"> <p>Optie A</p> </div> <div style="margin-bottom: 10px;"> <p>Optie B</p> </div> <div style="margin-bottom: 10px;"> <p>Optie C</p> </div> <div> <p>Optie D</p> </div>
---	--

Kies één van de volgende mogelijkheden:

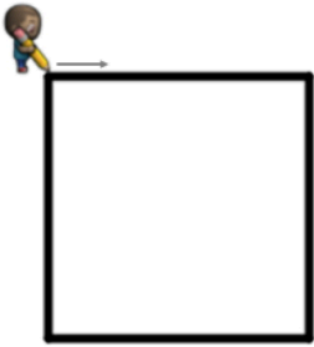
- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>De instructies moeten 'Pac-Man' naar de geest brengen via het gele pad. In welke stap van de instructies zit een <i>fout</i>?</p>	<div style="margin-bottom: 5px;"> <div style="background-color: #00a0c0; color: white; padding: 2px 5px; display: inline-block;">beweeg vooruit</div> → Stap A </div> <div style="margin-bottom: 5px;"> <div style="background-color: #00a0c0; color: white; padding: 2px 5px; display: inline-block;">draai links</div> <div style="background-color: #00a0c0; color: white; padding: 2px 5px; display: inline-block;">↺</div> → Stap B </div> <div style="margin-bottom: 5px;"> <div style="background-color: #00a0c0; color: white; padding: 2px 5px; display: inline-block;">beweeg vooruit</div> → Stap C </div> <div style="margin-bottom: 5px;"> <div style="background-color: #00a0c0; color: white; padding: 2px 5px; display: inline-block;">beweeg vooruit</div> → Stap D </div> <div style="margin-bottom: 5px;"> <div style="background-color: #00a0c0; color: white; padding: 2px 5px; display: inline-block;">draai links</div> <div style="background-color: #00a0c0; color: white; padding: 2px 5px; display: inline-block;">↺</div> → Stap D </div> <div style="margin-bottom: 5px;"> <div style="background-color: #00a0c0; color: white; padding: 2px 5px; display: inline-block;">beweeg vooruit</div> → Stap D </div>
--	---

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

Welke instructies moet de artiest volgen om het vierkant te tekenen? Alle kanten van het vierkant zijn 100 pixels lang.

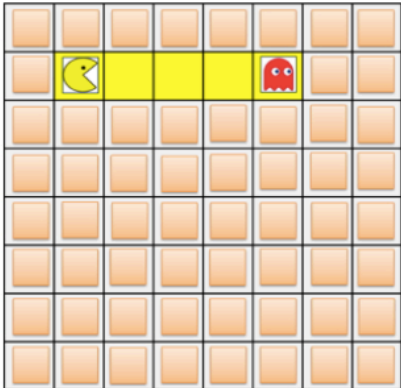






Optie A beweeg vooruit met 100 pixels draai rechts met 90 graden beweeg vooruit met 100 pixels draai rechts met 90 graden beweeg vooruit met 100 pixels draai rechts met 90 graden beweeg vooruit met 100 pixels	Optie B beweeg vooruit met 50 pixels draai rechts met 90 graden beweeg vooruit met 50 pixels draai rechts met 90 graden beweeg vooruit met 50 pixels draai rechts met 90 graden beweeg vooruit met 50 pixels
Optie C beweeg vooruit met 25 pixels draai rechts met 90 graden beweeg vooruit met 25 pixels draai links met 90 graden beweeg vooruit met 25 pixels draai rechts met 90 graden beweeg vooruit met 25 pixels	Optie D beweeg vooruit met 100 pixels draai rechts met 90 graden beweeg vooruit met 100 pixels draai links met 90 graden beweeg vooruit met 100 pixels draai rechts met 90 graden beweeg vooruit met 100 pixels

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D


Welke instructies brengen 'Pac-Man' naar de geest via het gele pad?



Optie A 	Optie B 
Optie C 	Optie D 

Kies één van de volgende mogelijkheden:

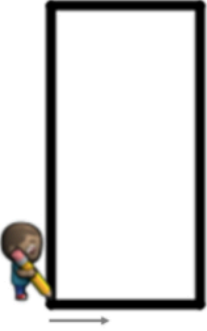
- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>Hoe vaak moet de <u>sequence</u> worden herhaald om 'Pac-Man' naar de geest te brengen via het gele pad?</p> 	<p>Optie A</p> <p>X 3</p> <hr/> <p>Optie B</p> <p>X 4</p> <hr/> <p>Optie C</p> <p>X 1</p> <hr/> <p>Optie D</p> <p>X 2</p>
---	---

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

De instructies moeten de artiest de volgende rechthoek 1 keer laten tekenen (50 pixels breed en 100 pixels hoog). In welke stap van de instructies zit een *fout*?



Stap A

```

herhaal 4 keer
doe
  beweeg vooruit met 50 pixels
  draai links met 90 graden
  beweeg vooruit met 100 pixels
  draai links met 90 graden

```

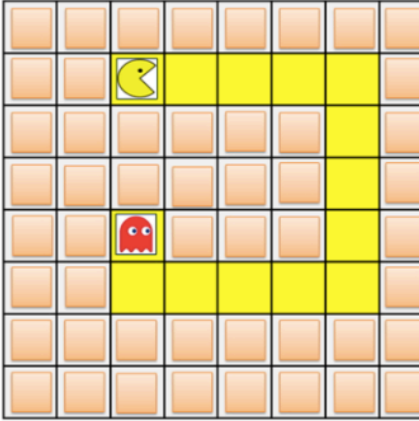
→ **Stap B**

→ **Stap C**

→ **Stap D**

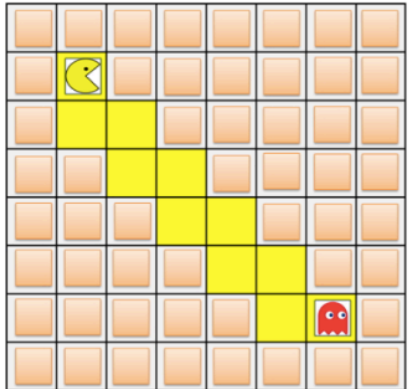












Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>Welke instructies brengen 'Pac-Man' naar de geest via het gele pad?</p> 	<p>Optie A</p> <pre> herhaal 4 keer doe beweeg vooruit herhaal 3 keer doe draai rechts beweeg vooruit </pre>	<p>Optie B</p> <pre> herhaal 3 keer doe herhaal 4 keer doe beweeg vooruit doe draai rechts beweeg vooruit </pre>
	<p>Optie C</p> <pre> herhaal 3 keer doe herhaal 4 keer doe beweeg vooruit doe draai rechts beweeg vooruit </pre>	<p>Optie D</p> <pre> herhaal 4 keer doe herhaal 3 keer doe beweeg vooruit doe draai rechts beweeg vooruit </pre>





Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>Welke instructies brengen 'Pac-Man' naar de geest via het gele pad?</p> 	<p>Optie A</p> <p>Herhaal tot de...   </p>	<p>Optie B</p> <p>Herhaal tot de...   </p>
	<p>Optie C</p> <p>Herhaal tot de...   </p>	<p>Optie D</p> <p>Herhaal tot de...   </p>

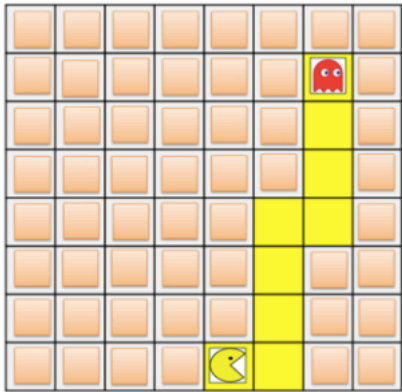
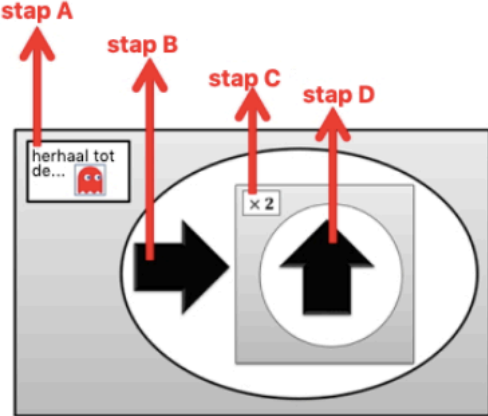
Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>Welke instructie mist om 'Pac-Man' naar de geest te brengen via het gele pad?</p> 	<p>Optie A</p> <p>Er mist geen stap</p>	<p>Optie B</p> 
	<p>Optie C</p> 	<p>Optie D</p> 

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>De instructies moeten 'Pac-Man' naar de geest brengen via het gele pad. In welke stap van de instructies zit een <i>fout</i>?</p> 	
--	--

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>Welke instructies brengen 'Pac-Man' naar de geest via het gele pad?</p>	<p>Optie A</p>	<p>Optie B</p>
	<p>Optie C</p>	<p>Optie D</p>

Kies één van de volgende mogelijkheden:

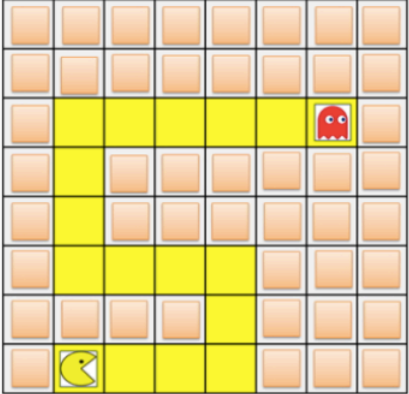
- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>Wat is de missende instructie om 'Pac-Man' naar de geest te brengen via het gele pad?</p>	<p>Optie A <i>Zowel optie A en optie C zijn goed</i></p> <p>Optie B </p> <p>Optie C </p> <p>Optie D </p>
--	--

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

De instructies moeten 'Pac-Man' naar de geest brengen via het gele pad. In welke stap van de instructies zit een *fout*?



```

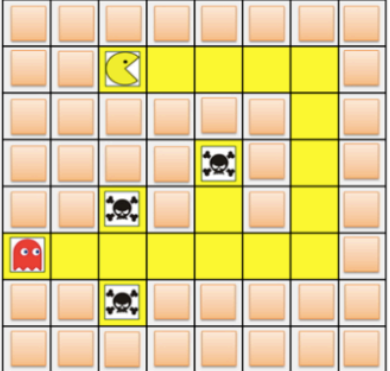
herhaal tot
doe
  beweeg vooruit
  als pad naar links
  doe draai links
  als pad naar rechts
  doe beweeg vooruit
  
```

→ Stap A
→ Stap B
→ Stap C
→ Stap D

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

Welke instructies brengen 'Pac-Man' naar de geest via het gele pad?



Optie A

```

herhaal tot
doe
  als pad naar links
  doe draai links
  anders beweeg vooruit
  
```

Optie B

```

herhaal tot
doe
  als pad naar rechts
  doe draai rechts
  anders beweeg vooruit
  
```

Optie C

```

herhaal tot
doe
  als pad vooruit
  doe beweeg vooruit
  anders draai rechts
  
```

Optie D

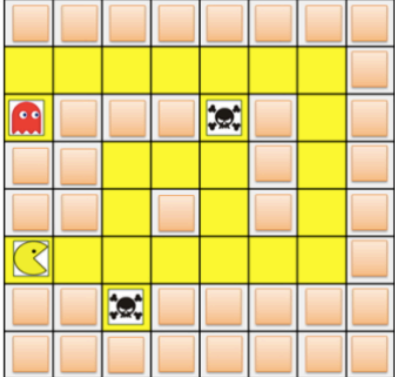
```

herhaal tot
doe
  als pad vooruit
  doe beweeg vooruit
  anders draai links
  
```

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

Welke instructies brengen 'Pac-Man' naar de geest via het gele pad?



Optie A

```

herhaal tot
doe
  als pad naar links
  doe draai links
  anders beweeg vooruit
        
```

Optie B

```

herhaal tot
doe
  als pad naar rechts
  doe draai rechts
  anders beweeg vooruit
        
```

Optie C

```

herhaal tot
doe
  als pad vooruit
  doe beweeg vooruit
  anders draai rechts
        
```

Optie D

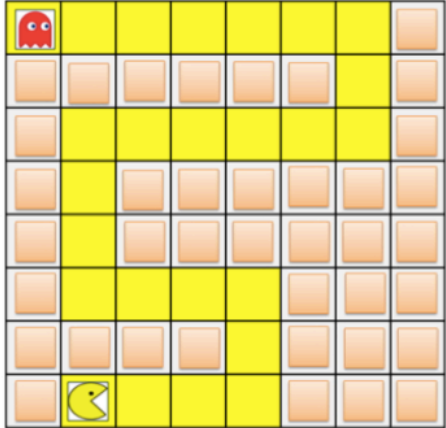
```

herhaal tot
doe
  als pad vooruit
  doe beweeg vooruit
  anders draai links
        
```

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

De instructies moeten 'Pac-Man' naar de geest brengen via het gele pad. In welke stap van de instructies zit een *fout*?



Herhaal tot

```

doe
  als pad vervolgt
  doe beweeg vooruit
  anders
    als pad naar rechts
    doe draai links
    anders draai rechts
        
```

→ Stap A





→ Stap B

→ Stap C

→ Stap D

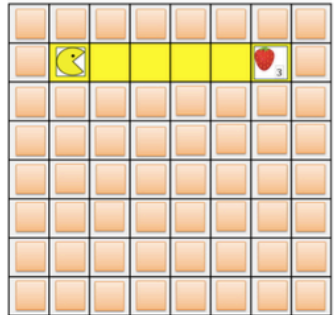




Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>Welke instructie mist om 'Pac-Man' naar de geest te brengen via het gele pad?</p> 	<p>Optie A</p> <p><i>Er mist geen stap</i></p>	<p>Optie B</p> 
	<p>Optie C</p> 	<p>Optie D</p> 

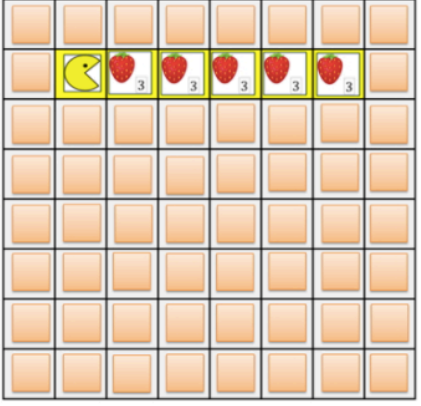
Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>Welke instructies brengen 'Pac-Man' naar de aardbeien via het gele pad en vertel 'Pac-Man' om alle getoonde aardbeien te eten?</p> 	<p>Optie A</p> 	<p>Optie B</p> 
	<p>Optie C</p> 	<p>Optie D</p> 

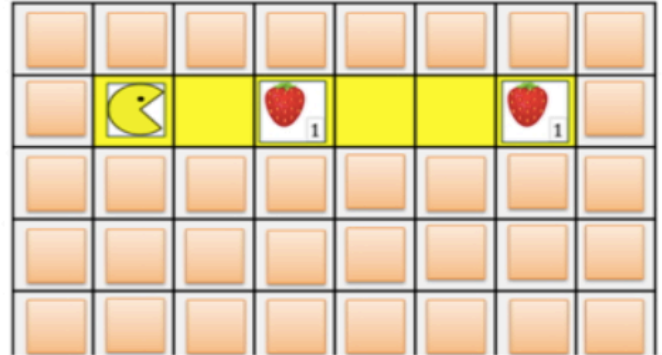
Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>Welke instructies brengen 'Pac-Man' naar de aardbeien via het gele pad en vertel 'Pac-Man' om alle getoonde aardbeien te eten?</p> 	<p>Optie A</p> <pre> als pad vooruit doe beweeg vooruit herhaal 3 keer doe Eet 1 aardbei </pre>	<p>Optie B</p> <pre> als pad vooruit doe herhaal 3 keer doe beweeg vooruit herhaal 5 keer doe Eet 1 aardbei </pre>
	<p>Optie C</p> <pre> als pad vooruit doe beweeg vooruit herhaal 3 keer doe Eet 1 aardbei </pre>	<p>Optie D</p> <pre> als pad vooruit doe herhaal 5 keer doe beweeg vooruit herhaal 3 keer doe Eet 1 aardbei </pre>

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>Wat mist er in de instructies hieronder om 'Pac-Man' naar de aardbeien te brengen via het gele pad en vertel 'Pac-Man' om alle getoonde aardbeien te eten?</p> <pre> als pad vooruit doe herhaal ?????????? doe beweeg vooruit als er aardbeien zijn doe Eet 1 aardbei </pre> 	<p>Optie A</p> <p><i>5 keer</i></p> <p>Optie B</p> <p><i>3 keer</i></p> <p>Optie C</p> <p><i>2 keer</i></p> <p>Optie D</p> <p><i>1 keer</i></p>
--	---

Kies één van de volgende mogelijkheden:

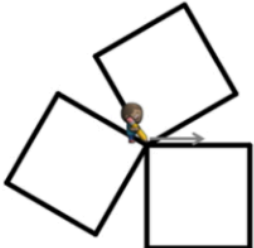
- ☐ A
- ☐ B

- C
- D

<p>Wat mist er in de instructies hieronder om 'Pac-Man' naar de aardbeien te brengen via het gele pad en vertel 'Pac-Man' om alle getoonde aardbeien te eten (aantal onbekend)?</p> <pre> als pad vooruit doe beweeg vooruit als een aardbei doe ????????????????? doe Eet 1 aardbei </pre> 	<p>Optie A</p> <p>Wanneer geen aardbeien</p> <p>Optie B</p> <p>wanneer aardbeien</p> <p>Optie C</p> <p>wanneer geen pad vooruit</p> <p>Optie D</p> <p>wanneer pad vooruit</p>
---	---

Kies één van de volgende mogelijkheden:

- A
- B
- C
- D

<p>De volgende de set van instructies welke 'mijn functie' worden genoemd, tekent een vierkant van 100 pixels aan elke kant:</p> <pre> functie mijn functie herhaal 4 keer doe beweeg vooruit met 100 pixels draai rechts met 90 graden </pre> <p>Welke instructies zou de artiest moeten volgen om het volgende ontwerp te tekenen?</p> <p>Elke kant van elk vierkant is 100 pixels lang.</p> 	<p>Optie A</p> <pre> herhaal 4 keer doe mijn functie draai rechts met 90 graden </pre> <p>Optie C</p> <pre> herhaal 3 keer doe mijn functie draai rechts met 120 graden </pre>	<p>Optie B</p> <pre> herhaal 3 keer doe mijn functie draai rechts met 90 graden </pre> <p>Optie D</p> <pre> herhaal 5 keer doe mijn functie draai rechts met 120 graden </pre>
--	--	--

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>De volgende set van instructies welke 'mijn functie' worden genoemd, tekent een driehoek van 50 pixels aan elke kant:</p> <div style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> <p>Functie</p> <pre> mijn functie herhaal 3 keer doe beweeg vooruit met 50 pixels draai links met 120 graden </pre> </div> <p>De onderstaande instructies moeten ervoor zorgen dat de artiest het volgende ontwerp tekent. Elke kant van de driehoek is 50 pixels lang. Wat is de missende instructie?</p> <div style="margin: 10px 0;"> <pre> herhaal ??? keer doe mijn functie spring vooruit met 50 pixels </pre> </div>	Optie A	Optie B	<div style="font-size: 2em; font-weight: bold;">3</div> <div style="font-size: 2em; font-weight: bold;">4</div>
	Optie C	Optie D	<div style="font-size: 2em; font-weight: bold;">5</div> <div style="font-size: 2em; font-weight: bold;">15</div>

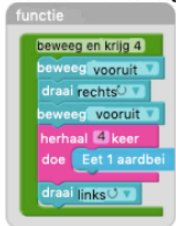
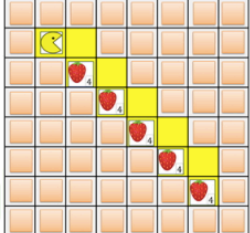
Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>De volgende set van instructies wordt 'krijg 5' genoemd:</p> <div style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> <p>Functie</p> <pre> krijg 5 herhaal 5 keer doe Eet 1 aardbei </pre> </div> <p>Welke instructies brengen 'Pac-Man' naar de aardbeien via het gele pad en vertellen 'Pac-Man' om alle getoonde aardbeien te eten?</p>	Optie A	Optie B	<div style="margin-bottom: 10px;"> <pre> beweeg vooruit draai rechts herhaal 5 keer doe krijg 5 beweeg vooruit </pre> </div> <div> <pre> beweeg vooruit draai rechts herhaal 3 keer doe krijg 5 beweeg vooruit </pre> </div>
	Optie C	Optie D	<div style="margin-bottom: 10px;"> <pre> beweeg vooruit draai rechts herhaal 5 keer doe beweeg vooruit doe krijg 5 </pre> </div> <div> <pre> beweeg vooruit draai rechts herhaal 3 keer doe beweeg vooruit doe krijg 5 </pre> </div>

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

<p>De volgende set van instructie wordt 'beweeg en krijg 4' genoemd:</p>  <p>Wat is de missende instructie hieronder om 'Pac-Man' naar de aardbeien via het gele pad te brengen en 'Pac-Man' te vertellen om alle getoonde aardbeien te eten?</p>  <p>herhaal ??? keer doe beweeg en krijg 4</p>	<p>Optie A</p> <p>6</p>	<p>Optie B</p> <p>5</p>
	<p>Optie C</p> <p>4</p>	<p>Optie D</p> <p>3</p>

Kies één van de volgende mogelijkheden:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

Je bent nu klaar, je mag nu rustig terug naar de klas.



Verzend uw enquête.





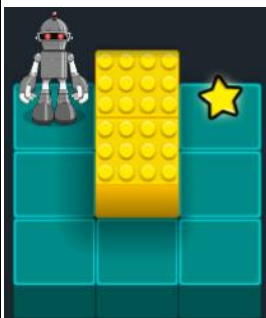

Bedankt voor uw deelname aan deze enquête.

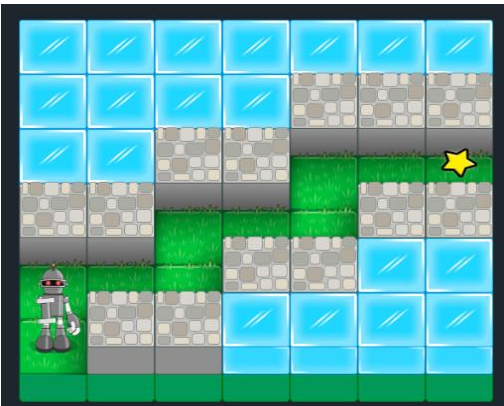

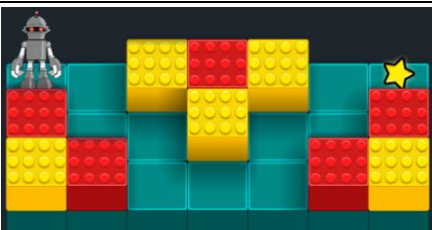



Bijlage 3: Interventiesessies

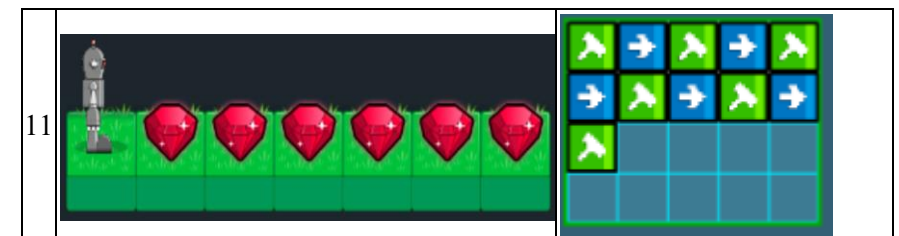
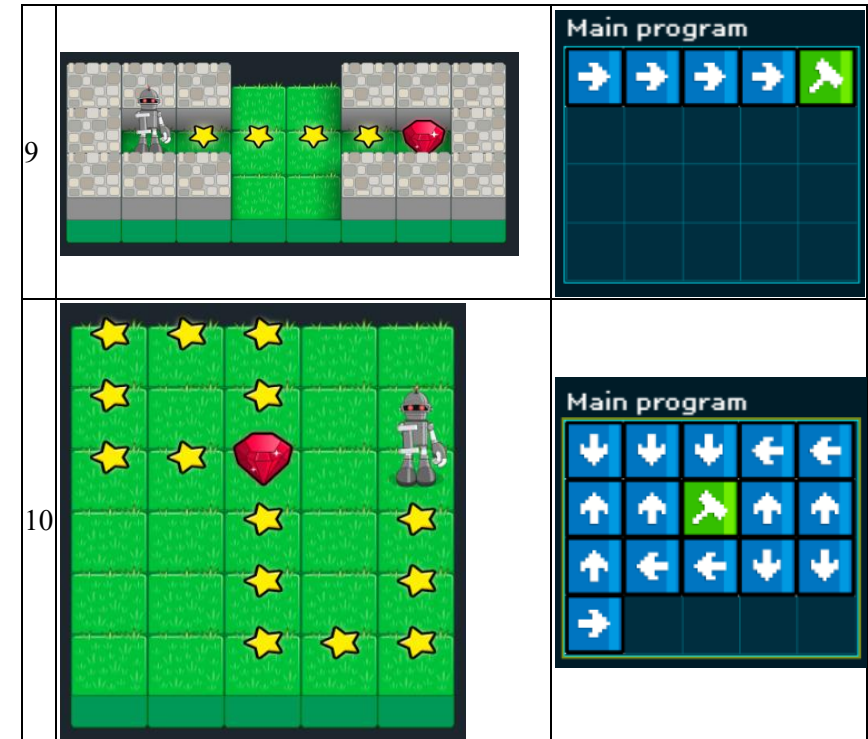
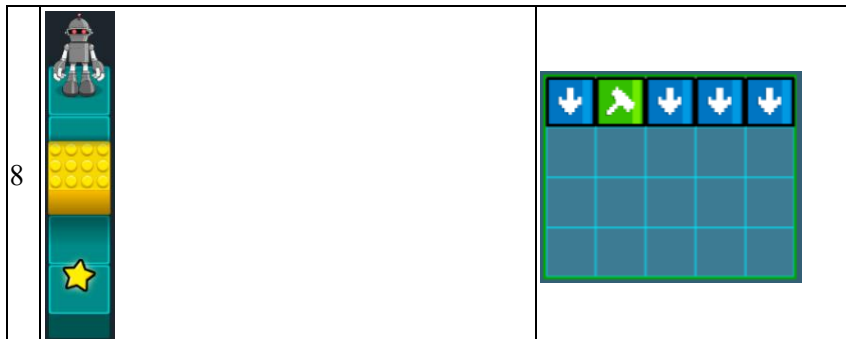
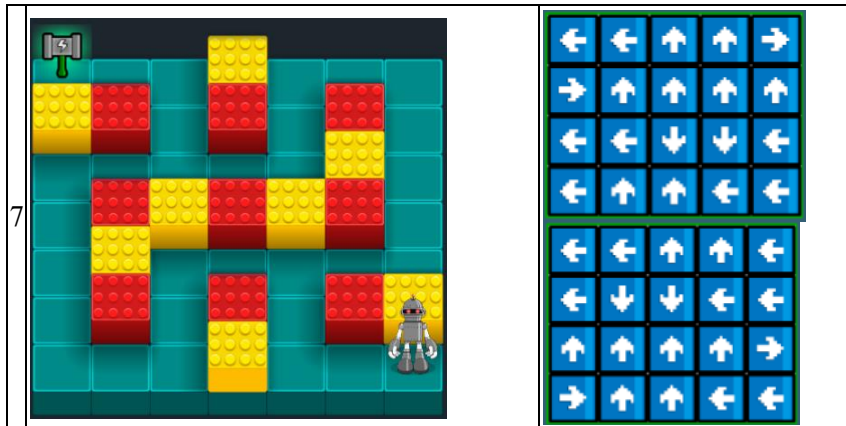
Sessie 1

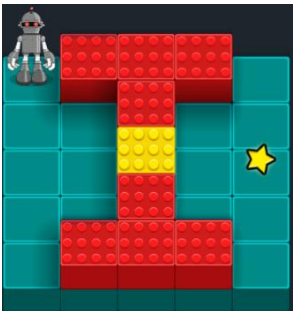

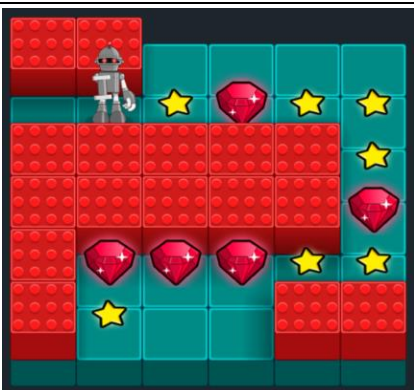

		Bomberbot	Lego EV3
Sessie 1	Doelen:	<ul style="list-style-type: none"> - Introductie begrip en programmeren - Leerlingen begrijpen dat een computerprogramma bestaat uit een set instructies die een computer zelf uitvoert en dat zij dit zelf ook kunnen maken. - Leerlingen kunnen simpele programma's maken met visuele programmeertaal - De leerlingen begrijpen het concept 'for loops' en identificeren patronen om levels efficiënter op te lossen - De leerlingen passen d.m.v. "for loops" herhaling toe om tot een betere oplossing voor problemen te komen. 	<ul style="list-style-type: none"> - Introductie begrip en programmeren - Leerlingen begrijpen dat een computerprogramma bestaat uit een set instructies die een computer zelf uitvoert en dat zij dit zelf ook kunnen maken. - Leerlingen kunnen simpele programma's maken met visuele programmeertaal - De leerlingen begrijpen het concept 'for loops' en identificeren patronen om levels efficiënter op te lossen - De leerlingen passen d.m.v. "for loops" herhaling toe om tot een betere oplossing voor problemen te komen.
	Lesplan:	<ul style="list-style-type: none"> - Introductie van begrippen en programma - Leg uit dat programmeren de manier is waarop mensen de computer opdrachten geven om iets uit te voeren. - Vertel: Om een computer opdrachten te kunnen geven, moet je met de computer communiceren in 'computertaal'. Net zoals er verschillende talen zijn (Engels, Frans, Spaans) zijn er ook veel verschillende programmeertalen en dus ook verschillende manieren om met computers te communiceren. Maar hoe verschillend of complex al deze programmeertalen ook zijn, uiteindelijk vertellen ze allemaal wat de computer moet doen. - Vertel: Jullie gaan leren programmeren door middel van een spel waarin jullie een speciale robot, genaamd Bomberbot, opdrachten geven. We geven de robot instructies om zich te verplaatsen en puzzels op te 	<ul style="list-style-type: none"> - Introductie van begrippen en programma - Leg uit dat programmeren de manier is waarop mensen de computer opdrachten geven om iets uit te voeren. - Vertel: Om een computer opdrachten te kunnen geven, moet je met de computer communiceren in 'computertaal'. Net zoals er verschillende talen zijn (Engels, Frans, Spaans) zijn er ook veel verschillende programmeertalen en dus ook verschillende manieren om met computers te communiceren. Maar hoe verschillend of complex al deze programmeertalen ook zijn, uiteindelijk vertellen ze allemaal wat de computer moet doen. - Vertel: Jullie gaan leren programmeren door middel van een spel waarin we een speciale robot opdrachten geven. Jullie geven de robot instructies om zich te

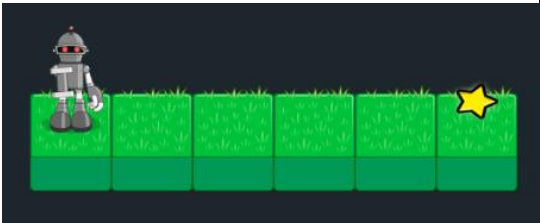

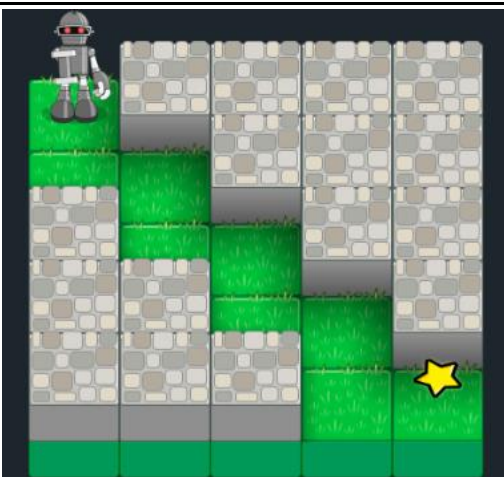

		<p>lossen. We moeten dit doen met zo min mogelijk instructies.</p> <p>Bomberbot weet pas dat hij het programma moet uitvoeren zodra je die opdracht geeft. Druk op de speel knop om de robot het programma uit te laten voeren. Klik opnieuw op deze knop om de robot te pauzeren. Klik op de terug knop om de robot terug te zetten naar de beginpositie. Dit kan handig zijn als je even niet meer weet wat je moet doen. In sommige levels kun je op de knop met het hoofd van Andrew (de robottrainer) drukken om het bericht nogmaals te lezen. In sommige levels kun je ook op een knop met vraagtekens klikken om hulp te krijgen.</p>	<p>verplaatsen en puzzels op te lossen. We moeten dit doen met zo min mogelijk instructies.</p> <p>De robot weet pas dat hij het programma moet uitvoeren zodra je die opdracht geeft. Hiervoor moet je het programma downloaden en uitvoeren. Wanneer je het programma verkeerd hebt geschreven kan je het programma gewoon veranderen of opnieuw schrijven. Wanneer je dan weer op downloaden en uitvoeren klikt zal hij het nieuwe programma gaan uitvoeren.</p>
	Uitvoeren:	De leerlingen gaan het programma ontdekken en krijgen hierbij de volgende missie: level 1-13 uit van les 1. Daarna gaan ze verder met les 3 (zie antwoorden voor levels).	Leerlingen gaan het programma ontdekken en krijgen hierbij de volgende missie: De leerlingen krijgen de missie om de robot een afstand van precies 10 cm. Rechtdoor te laten rijden.
	Vertel:	<p>Om een set instructies een bepaald aantal keer te herhalen gebruik je binnen Bomberbot de volgende instructies:</p> 	<p>Om een set instructies een bepaald aantal keer te herhalen kan je hiervoor binnen het programmeerprogramma het volgende blok:</p> <p>Je kan hier ook het aantal herhalingen aanpassen</p> 
	Uitvoeren	Leerlingen voeren level 1-10 uit van les 3. Daarna zijn ze klaar met de eerste sessie.	<p>Missie uitvoeren. De leerlingen krijgen de missie om de robot een afstand van precies 5 cm rechtdoor te laten rijden en dit 3 keer te herhalen.</p> <p>Wanneer de leerlingen klaar zijn met deze missie zijn ze klaar met deze sessie.</p>

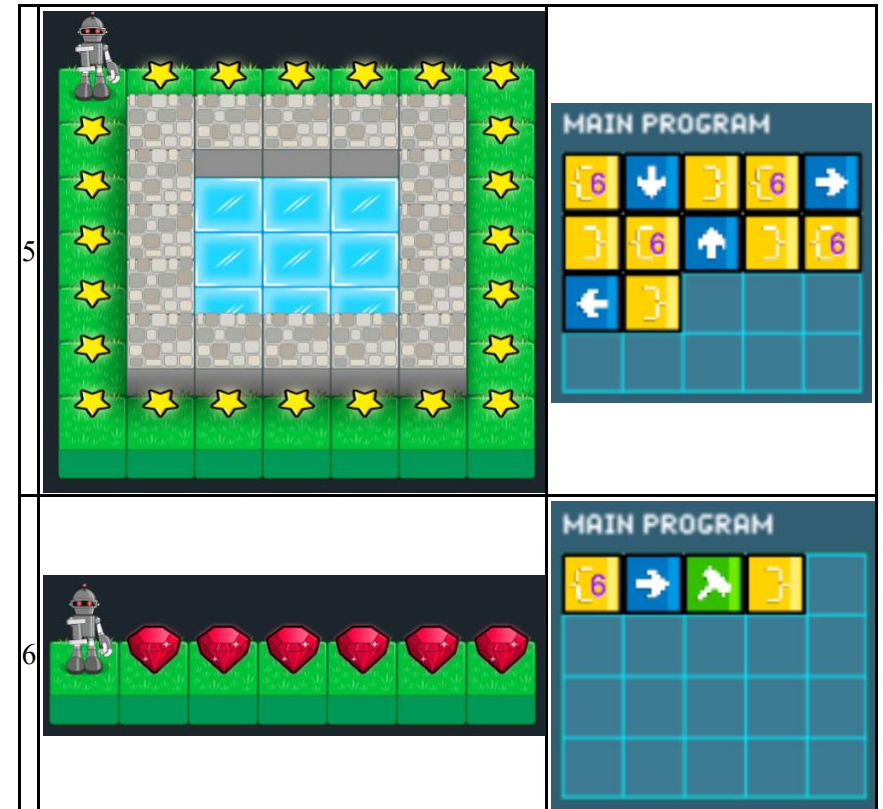
#	Deel 1 Bomberbot	Beste oplossing
1		
2		
3		

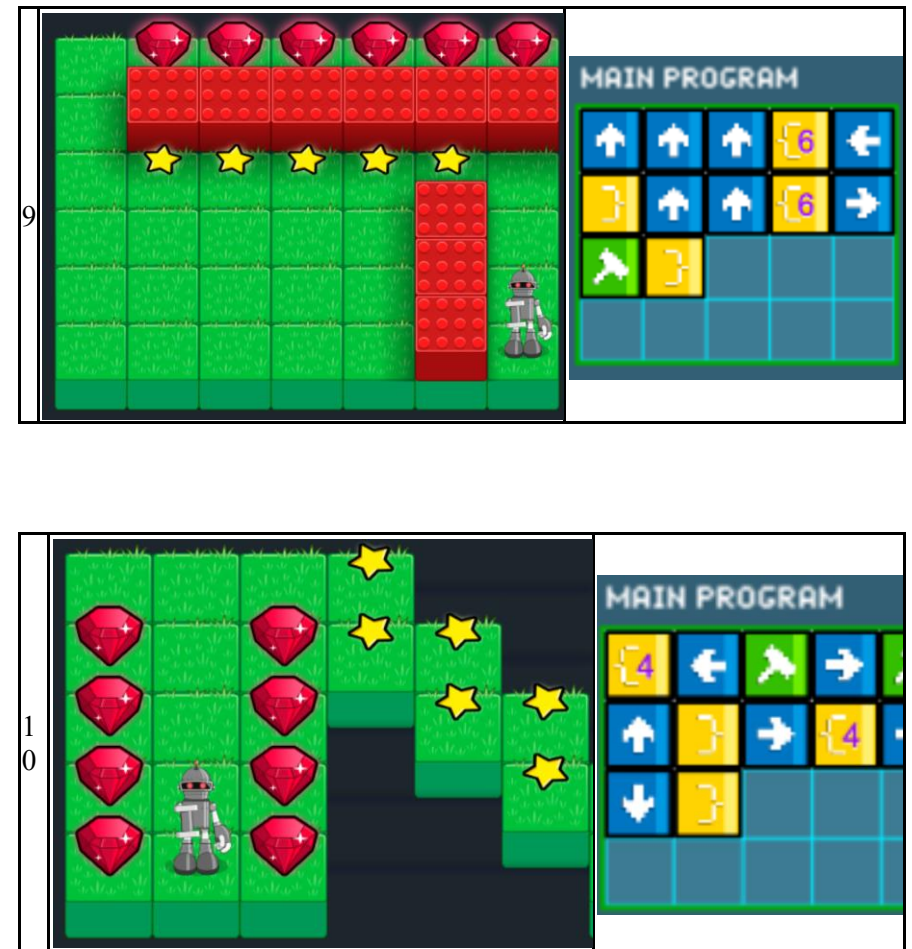
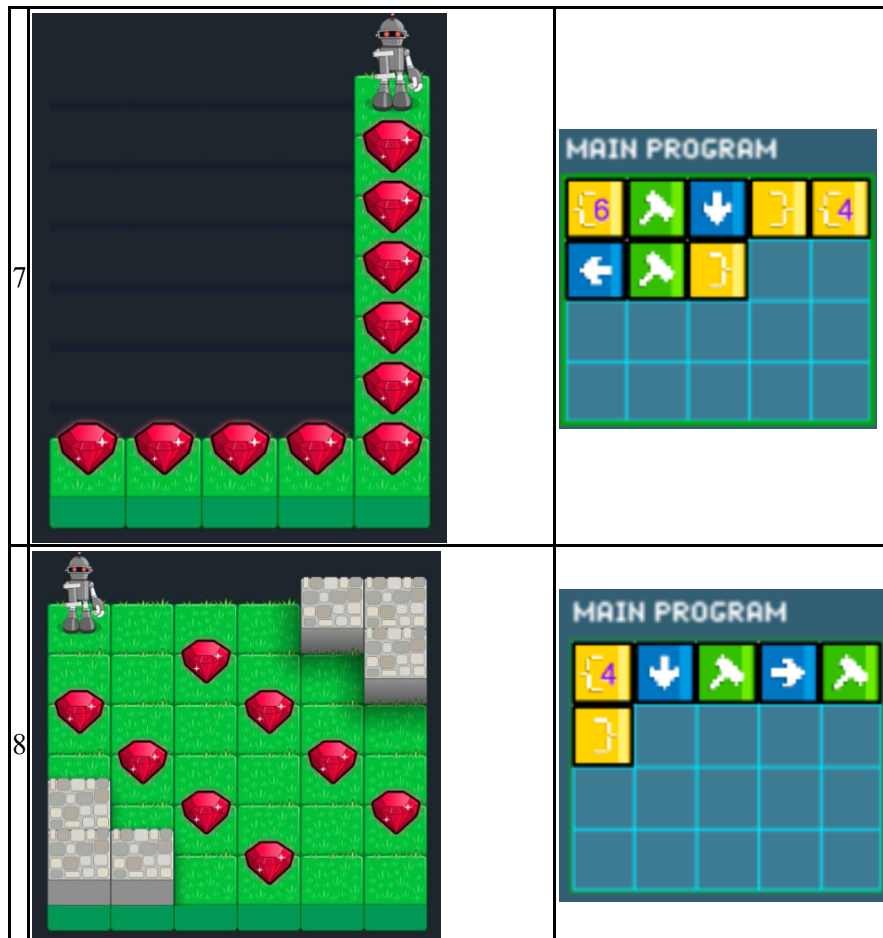
4		
5		
6		



12		
13		

Deel 2		
1		
2		

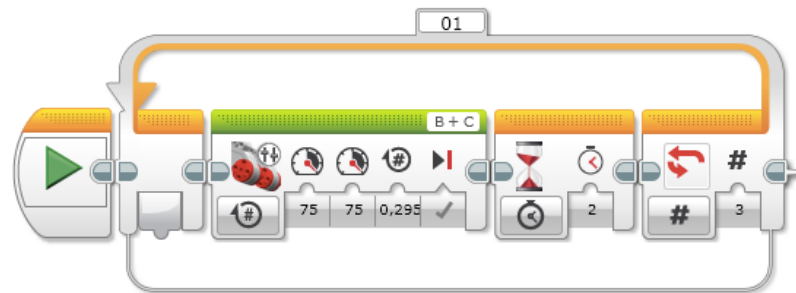




Oplossing sessie 1 LEGO ev3: 10 cm vooruit


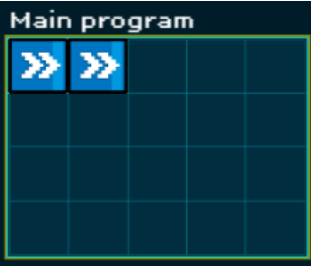
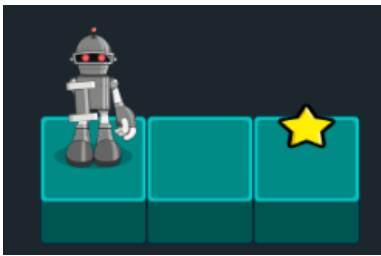
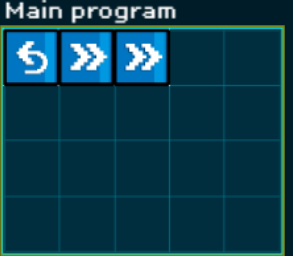
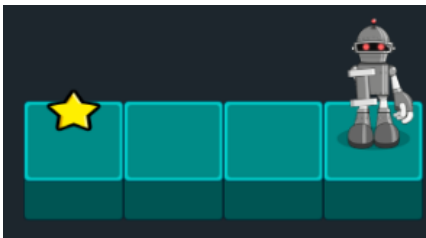
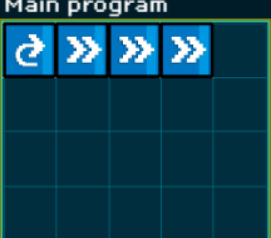


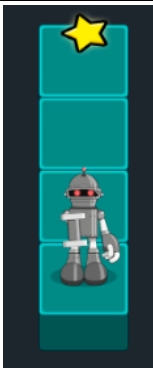
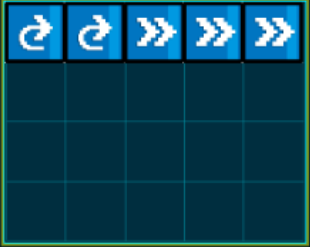
Oplossing sessie 1 LEGO ev3: 5 cm 3 keer herhalen. (Kan ook zonder blok zandloper).



Sessie 2

		Bomberbot	Lego EV3
Sessie 2	Doelen:	<ul style="list-style-type: none"> - De leerlingen oefenen met ruimtelijk redeneren en relatieve richtingen - De leerlingen begrijpen hoe elke opdracht of instructie de positie van de robot en de gehele sequentie beïnvloedt. 	<ul style="list-style-type: none"> - De leerlingen oefenen met ruimtelijk redeneren en relatieve richtingen - De leerlingen begrijpen hoe elke opdracht of instructie de positie van de robot en de gehele sequentie beïnvloedt.
	Lesplan:	Jullie gaan de robot nu sturen door de robot te laten draaien. Je krijgt er dan nog een andere knop bij waarmee je de robot rechtdoor laat lopen. Dit hangt dus af van welke kant de robot op kijkt.	<p>Jullie gaan de robot nu sturen door de robot te laten draaien. Missie: (vierkant van 15 cm bij 15 cm). Er zit een stift aan je robot geplakt en deze moet precies het vierkant overtrekken wat op de grond is geplakt. (Vierkant van 15 cm bij 15 cm).</p> <p>Wanneer leerlingen eerder klaar zijn: Nu zorg je dat je het vierkant en het kruis wat er in staat tekent.</p> <p>*let op: ALERT OP AANTAL GRADEN. gebruik stuurblok: wielkje. Instellen rotaties/graden aan. Draai scherpste op max. (schuifbalkje). Verdubbelen van 90 graden.</p>
	Uitvoeren:	De leerlingen krijgen de volgende missie: level 1-12 uit van les 4.	<p>Missie: (vierkant van 15 cm bij 15 cm). Er zit een potlood aan je robot geplakt en deze moet precies het vierkant overtrekken wat op de grond is geplakt. (Vierkant van 15 cm bij 15 cm).</p> <p>Wanneer leerlingen eerder klaar zijn: Nu zorg je dat je het vierkant en het kruis wat er in staat tekent.</p>

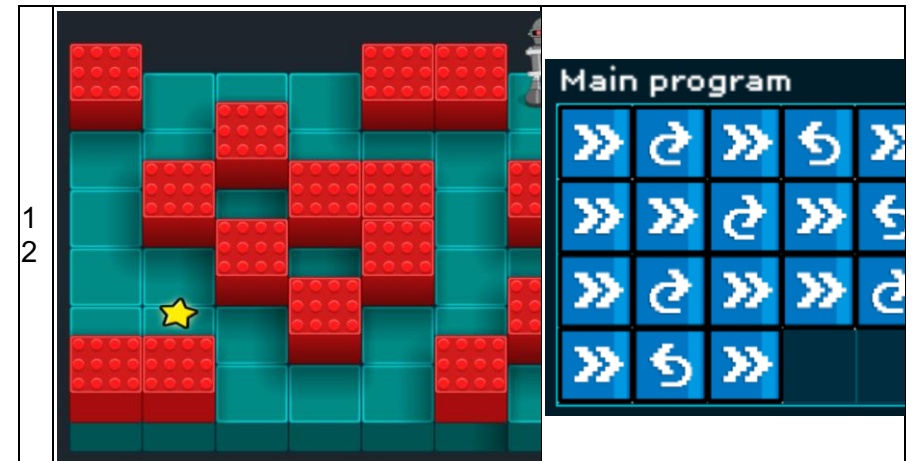
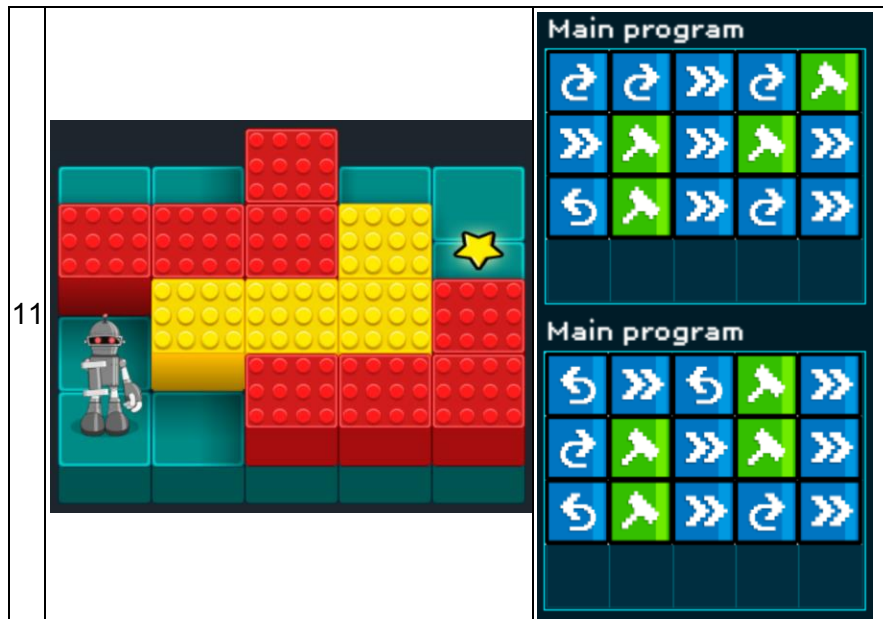
Sessie 2	Beste oplossing
1 	Main program 
2 	Main program 
3 	Main program 

4 	Main program 
---	---

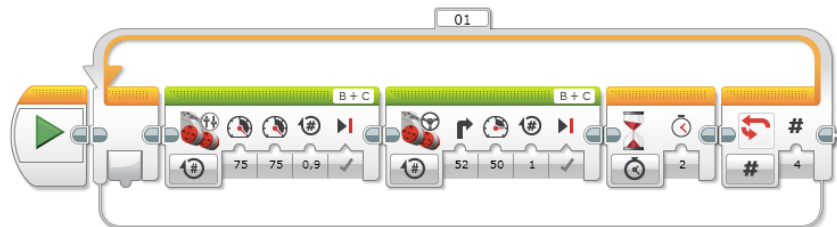
5 	Main program 
--	--

6		Main program >> ↻ >> >> ↶ >> >> ↻ >> >>
7		Main program ↻ >> ↶ ↻ >> ↻
8		Main program >> >> ↻ >> ↶ >> >> ↻

9		Main program >> ↻ >> ↻ >> >> ↻ >> >> ↻ >> >>
10		Main program ↻ ↻ ↻ ↻ ↻ ↻ >> ↻ ↻ ↻ >> >> ↻



Oplossing sessie 2 Lego Ev3:
1. Vierkant 15x 15



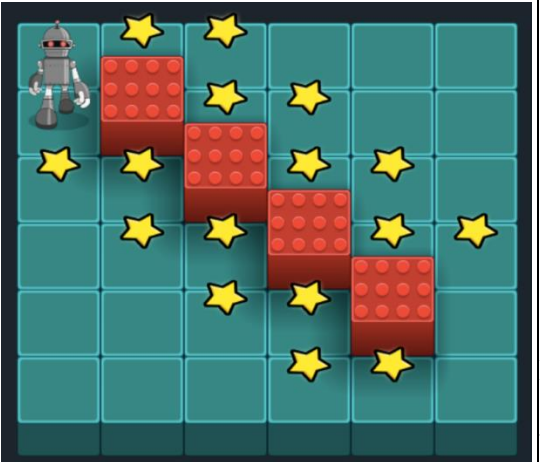



		Bomberbot	Lego EV3
Sessie 3	Doelen:	<ul style="list-style-type: none"> - De leerlingen kunnen simpele algoritmes met for loops uitschrijven om een probleem op te lossen. - De leerlingen kunnen voorbeelden van algoritmes uit het dagelijks leven identificeren. 	<ul style="list-style-type: none"> - De leerlingen oefenen met ruimtelijk redeneren en relatieve richtingen - De leerlingen begrijpen hoe elke opdracht of instructie de positie van de robot en de gehele sequentie beïnvloedt.
	Lesplan:	<p>Vertel: Stel we willen een algoritme schrijven om pannenkoeken te maken voor vier personen. Hoe ziet dit er dan uit? Op het bord staat het algoritme voor 1 persoon.</p> <p>Antwoord:</p> <ol style="list-style-type: none"> 1. Meng ei en boter in een grote kom 2. Voeg bloem en suiker toe 3. Meng alle ingrediënten samen 4. Zet een pan op het fornuis en verhit deze <p>Herhaal vier keer: (dit moet worden toegevoegd door de leerlingen)</p> <ol style="list-style-type: none"> 5. Voeg een klein beetje olie toe 6. Voeg beslag toe in de pan 7. Draai de pannenkoek om 8. Leg de pannenkoek 	<p>Vertel: Stel we willen een algoritme schrijven om pannenkoeken te maken voor vier personen. Hoe ziet dit er dan uit? Op het bord staat het algoritme voor 1 persoon.</p> <p>Antwoord:</p> <ol style="list-style-type: none"> 1. Meng ei en boter in een grote kom 2. Voeg bloem en suiker toe 3. Meng alle ingrediënten samen 4. Zet een pan op het fornuis en verhit deze <p>Herhaal vier keer: (dit moet worden toegevoegd door de leerlingen)</p> <ol style="list-style-type: none"> 5. Voeg een klein beetje olie toe 6. Voeg beslag toe in de pan 7. Draai de pannenkoek om

Sessie

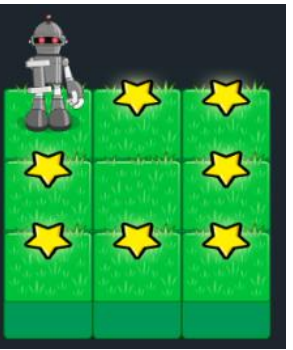


Antwoorden Bomberbot:

Sessie 3:

1		
2		

Er is een betere oplossing (hieronder), maar we verwachten nog niet dat

	<p>leerlingen deze op dit moment vinden.</p> <div> <p>MAIN PROGRAM</p> <table border="1"> <tr> <td>{5</td> <td>↓</td> <td>→</td> <td>}</td> <td>{5</td> </tr> <tr> <td>↑</td> <td>←</td> <td>}</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table> </div>	{5	↓	→	}	{5	↑	←	}												
{5	↓	→	}	{5																	
↑	←	}																			
3	 <div> <p>MAIN PROGRAM</p> <table border="1"> <tr> <td>{4</td> <td>>></td> <td>↶</td> <td>>></td> <td>↷</td> </tr> <tr> <td>}</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table> </div>	{4	>>	↶	>>	↷	}														
{4	>>	↶	>>	↷																	
}																					

4	 <div> <p>MAIN PROGRAM</p> <table border="1"> <tr> <td>{4</td> <td>>></td> <td>>></td> <td>↶</td> <td>}</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table> </div>	{4	>>	>>	↶	}															
{4	>>	>>	↶	}																	
5	 <div> <p>MAIN PROGRAM</p> <table border="1"> <tr> <td>{4</td> <td>↖</td> <td>↷</td> <td>}</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </table> </div>	{4	↖	↷	}																
{4	↖	↷	}																		
6	 <div> <p>MAIN PROGRAM</p> <table border="1"> <tr> <td>{4</td> <td>↖</td> <td>↷</td> <td>}</td> <td>{3</td> </tr> <tr> <td>>></td> <td>>></td> <td>↷</td> <td>}</td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table> </div>	{4	↖	↷	}	{3	>>	>>	↷	}											
{4	↖	↷	}	{3																	
>>	>>	↷	}																		

7

MAIN PROGRAM

{3	↖	>>	↻	>>
>>	↶	}		

9

MAIN PROGRAM

{3	↖	>>	↖	>>
↶	}	{2	→	↑
→	↓	↓	}	

8

MAIN PROGRAM

{3	>>	>>	↶	}
>>	↑	↑	{4	>>
>>	↻	}		

MAIN PROGRAM

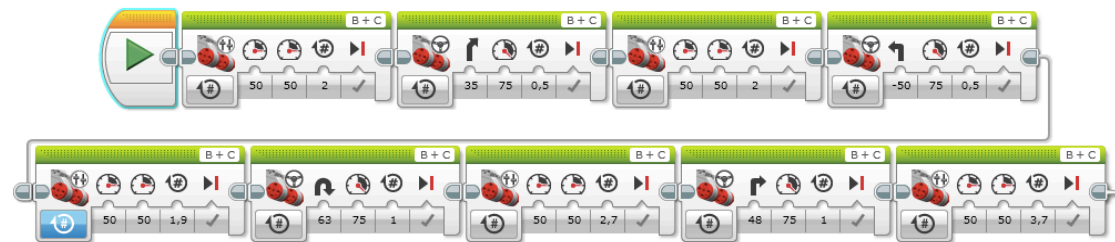
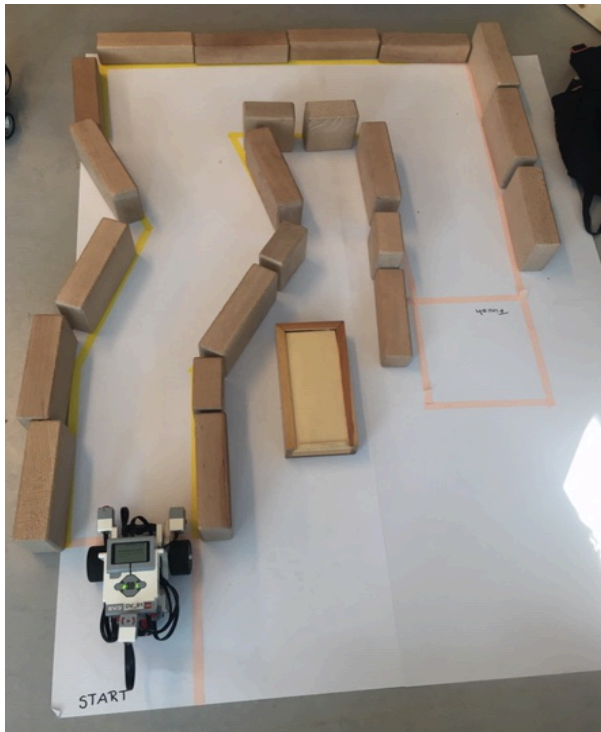
{4	>>	>>	↶	}
↑	↑	→	{4	>>
>>	↶	}		

10

MAIN PROGRAM

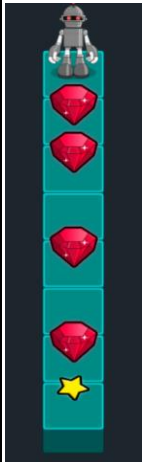
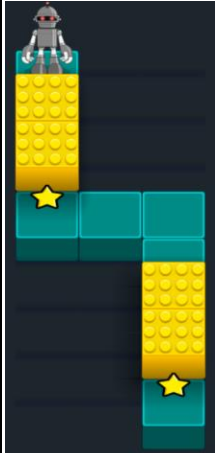
{6	→	↖	}	↑
↖	↑	{4	↖	>>
↖	>>	↶	}	

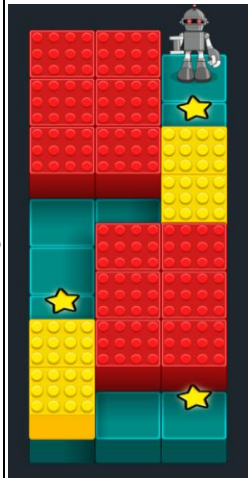
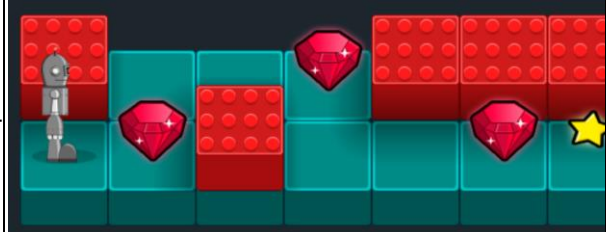
Opstelling/oplossing sessie 3 Lego EV3



Sessie 4

		Bomberbot	Lego EV3
Sessie 4	Doelen:	<ul style="list-style-type: none"> - Leerlingen stappen het begrip van een opgeslagen functie. Een vooraf gestelde set van instructies. - Leerlingen kunnen problemen opdelen in sub-problemen. 	<ul style="list-style-type: none"> - Leerlingen stappen het begrip van een opgeslagen functie. Een vooraf gestelde set van instructies. - Leerlingen kunnen problemen opdelen in sub-problemen.
	Lesplan:	Vertel: een sub-sequentie van instructies die opgeslagen en hergebruikt kan worden in een programma; een routine die een programma opdeelt in kleinere delen. Wanneer je steeds hetzelfde moet doen, kan je je voorprogrammeren met een functie. Denk maar aan de programma's van de wasmachine of de vaatwasser.	<p>Vertel: een sub-sequentie van instructies die opgeslagen en hergebruikt kan worden in een programma; een routine die een programma opdeelt in kleinere delen. Wanneer je steeds hetzelfde moet doen, kan je je voorprogrammeren met een functie. Denk maar aan de programma's van de wasmachine of de vaatwasser.</p> <p>En zou je dit handiger kunnen doen? (Denk aan fabrieken) Wanneer leerlingen niet op het toevoegen van een sensor komen, dit vanuit de begeleider initiëren.</p>
	Uitvoeren:	De leerlingen krijgen de volgende missie: level 1-12 uit van les 6.	Missie: Je stuurt je robot door het doolhof wat op de grond is neergezet. De blokken mogen niet verplaatsen. In dit doolhof zal 3 keer dezelfde manoeuvre moeten worden uitgevoerd. Waardoor de leerlingen 3 keer dezelfde set instructies (functie) kunnen programmeren. In deze set instructies hebben de leerlingen de mogelijkheid om gebruik te maken van de kleurensensor.

Sessie 4 Bomberbot		Beste oplossing
1		<div> Main program F1 F1 ↓ F1 ↓ F1 ↓ </div> <div> Function 1 (F1) ↖ ↓ </div>
2		<div> Main program F1 → → ↓ F1 </div> <div> Function 1 (F1) ↖ ↓ ↖ ↓ ↓ </div>

3		<div> Main program F1 ← ← ↓ F1 → → </div> <div> Function 1 (F1) ↓ ↖ ↓ ↖ ↓ </div>
4		<div> Main program F1 ↑ → F1 ↓ → F1 → </div> <div> Function 1 (F1) ↖ → </div>

5

Main program

F1

↻

➡➡

F1

Function 1 (F1)

⬆

➡➡➡➡

6

Main program

F1 F1 ➡➡ ↶ F1

F1

Function 1 (F1)

⬆ ➡

Main program

F1 ➡➡ ↶ F1

Function 1 (F1)

⬆ ➡ ⬆ ➡

7

Main program

F1 ➡➡ F1 F1

Function 1 (F1)

➡➡ ↻

8

Main program

F1 ➡ ➡ ➡ ➡

➡ ➡ ➡ F1

Function 1 (F1)

⬆ ⬆ ⬆ ⬆

⬆

Main program

F1 F1 F1 ➡ ➡

➡ ➡ ➡ ➡

F1 F1 F1

Function 1 (F1)

⬆ ⬆

9

Main program

↓	F1	↓	↓	↓
↓	F1	↑	↑	F1

Function 1 (F1)

←	↖	←	↖	

10

Main program

F1	←	←	←	←
↑	F1			

Function 1 (F1)

←	←	↑	↑	→
→	↓			

Main program

F1	←	←	←	↑
↑	F1			

Function 1 (F1)

↑	↑	←	←	↓
↓	→			

11

Main program

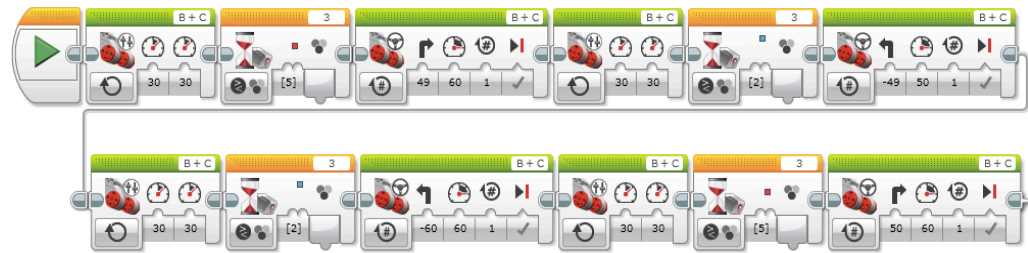
F1	←	←	←	↑
F1				

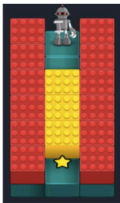
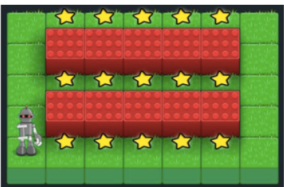


Function 1 (F1)

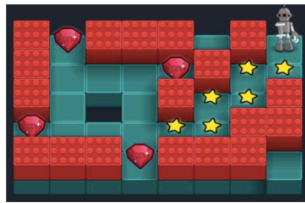
↖	↗	↖	↗	↻
↖	↗			

12

A photograph of a custom-built robot on a white paper track. The track is marked with black lines and contains two blue rectangular obstacles and two red rectangular obstacles. A 'Finish' line is marked at the bottom right. The robot is a small, white, custom-built vehicle with a camera and sensors, positioned at the top right of the track.

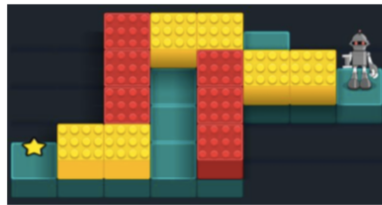
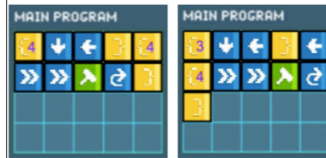


		Bomberbot	Lego EV3
Sessie 5			
	Lesplan:	Vertel: Jullie gaan een afsluitende challenge uitvoeren waarbij jullie steeds deel challenges moeten oplossen. Hiervoor heb je per deelchallenge 3 pogingen en 15 minuten de tijd. Zorg dat je het programmeren zo precies mogelijk doet. Gebruik geen onnodige blokken.	Vertel: Jullie gaan een afsluitende challenge uitvoeren waarbij jullie steeds deel challenges moeten oplossen. Hiervoor heb je per deelchallenge 3 pogingen en 15 minuten de tijd. Zorg dat je het programmeren zo precies mogelijk doet. Gebruik geen onnodige blokken.
	Uitvoeren:	<p>De leerlingen moeten de volgende missies oplossen: Waarbij je alle sterren en robijnen verzamelt met een zo kort mogelijk programma.</p> <div>  <p>1: (*les 1 opgave 22)</p> </div> <div>  <p>2: (*les 3 opgave 19)</p> </div> <div>  </div> <div>  </div>	<p>De deelchallenges worden in het “doolhof” aangegeven door middel van onderbroken strepen.</p> <p>Deelchallenge 1: Je robot begint bij start rijdt in een rechte lijn en stopt precies op het punt van start 2. De robot mag de blokken niet omverrijden en aanraken.</p> <p>Deelchallenge 2: Je robot begint bij start 2 en rijdt door het doolhof en stopt precies bij start 3. (Nu zal de robot bochten moeten maken).</p> <p>Deelchallenge 3: Je robot begint bij start 3 en rijdt door het doolhof en stopt precies bij start 4.</p> <p>Deelchallenge 4: Je robot begint bij start 4 en rijdt door het doolhof en stopt precies bij start 5 (hierbij kan gebruik gemaakt worden van de kleurensensor).</p> <p>Deelchallenge 5: Je robot begint bij start 5 en rijdt door het doolhof naar de finish. Maak zo optimaal mogelijk gebruik van je robot. (sensor)</p>



7261

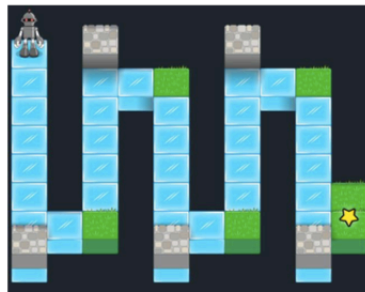
3:
(*les 5 opgave 19)



8357

4:
(*les 6 opgave 14)

5:



2680

(*les 8 opgave 13)

